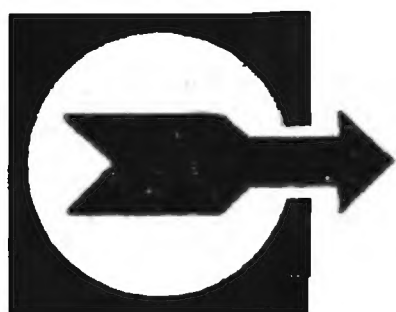


VISIT...

LANZAROTE
Caliente.COM

Л. А. Растрюгин



**СОВРЕМЕННЫЕ
ПРИНЦИПЫ
УПРАВЛЕНИЯ
СЛОЖНЫМИ
ОБЪЕКТАМИ**

ББК 32.815

Р 24

УДК 681.5→007

Растригин Л. А.

Р 24 Современные принципы управления сложными объектами. — М.: Сов. радио, 1980. — 232 с. ил.

В пер.: 90 к.

Книга является первым в мировой литературе опытом «сквозного» изложения современных принципов, подходов и методов управления сложными формализуемыми объектами различной физической природы — техническими, технологическими и т. д. Для ее чтения достаточно знания основ современной математики в объеме обычного курса вуза. Книга рассчитана на инженеров различных специальностей, желающих ознакомиться с методами теории и практики управления сложными объектами.

Р 30502-021
046(01)-80

49-79 2704080330

ББК 32.815

687.3

Редакция кибернетической литературы

ИБ № 512

ЛЕОНАРД АНДРЕЕВИЧ РАСТРИГИН

**Современные принципы управления
сложными объектами**

Научный редактор *Т. А. Колоскова*

Редактор *Т. М. Любимова*

Художник *В. В. Кухта*

Технический редактор *И. В. Орлова*

Корректор *Н. М. Давыдова*

Сдано в набор 26.09.79

Подписано в печать 27.12.79

T-21645

Формат 84×108¹/₃₂

Бумага типографская № 2

Гарнитура литературная

Печать высокая Объем 12,18 усл. п. л. 12,36 уч.-изд. л. Тираж 12 000 экз.

Зак. 299

Цена 90 к.

Издательство «Советское радио», Москва, Главпочтамт, в/я 693

Московская типография № 10 «Союзполиграфпрома»

Государственного Комитета СССР

по делам издательств, полиграфии и книжной торговли.

Москва, М-114, Шлюзовая наб., 10

© Издательство «Советское радио», 1980 г.

*Знание некоторых принципов нередко
возмещает незнание некоторых фак-
тов.*

ГЕЛЬВЕЦИИ

Характерным для нашего времени является стремление использовать формальные (математические) методы для управления сложными объектами. На этом пути сделано уже достаточно много. Однако при синтезе процесса управления сложной системой неизбежно также использование неформальных (экспертных) методов. Это обстоятельство нельзя рассматривать как «заметание мусора под коврик». Дело в том, что эксперт является таким же источником информации, как любой датчик, и, как всякий датчик, обладает своими особенностями, которые необходимо учитывать при его использовании. Сочетание формальных и неформальных методов и является спецификой управления сложным объектом, которая отражена в этой книге.

В книге рассмотрены все этапы управления сложным объектом — от формулировки целей управления до коррекции этих целей в процессе эксплуатации созданной системы управления.

Процесс создания управления в книге рассмотрен с алгоритмической точки зрения, независимо от того, с помощью каких вычислительных средств эти алгоритмы будут реализованы (и будут ли реализованы вообще). В этом плане книгу дополняют руководства по проектированию систем управления (см., например: В. Н. Захаров, Д. А. Поспелов и В. Е. Хазацкий. Системы управления. — М.: Энергия, 1977), в которых рассмотрены способы и средства реализации алгоритмов управления на современной вычислительной технике.

Автор поставил цель — в доступной для инженера форме рассмотреть все восемь этапов управления сложным объектом. Каждая глава посвящена одному из эта-

пов создания системы управления. Так, гл. 1 посвящена первому этапу всякого управления — формулировке цели управления, гл. 2 — второму этапу — выделению объекта управления из среды. Следующие три главы посвящены синтезу модели выбранного управления (гл. 3 — структурному синтезу модели, а гл. 4 и 5 — параметрическому пассивному (при идентификации) и активному (при экспериментировании с объектом). Синтез управления на базе созданной модели объекта рассмотрен в гл. 6. Гл. 7 посвящена этапу реализации управления. И, наконец, в последней главе рассматривается этап коррекции системы управления (модели, объекта и целей) на базе полученной в процессе управления информации.

В книге представлен и оригинальный материал. Так, например, первые две главы целиком отражают исследования автором процессов целеобразования и выделения объекта из среды. В остальных главах предложения автора «вклиниваются» в материал, известный специалистам.

Естественно, что объять необъятную проблему управления сложными объектами в восьми главах невозможно. Поэтому автор рассмотрел лишь те алгоритмы, методы и подходы теории и практики управления, которые наилучшим образом отражают современные принципы управления сложным объектом. При этом очевидно, что какие-то аспекты проблемы управления вовсе не рассмотрены. Так, совершенно не представлены в книге иерархические подходы к описанию и управлению сложными объектами — так называемый системный подход, модели АСУ и т. д. Основная, пожалуй, тому причина — ограниченный объем книги.

Было бы ошибочным считать эту книгу справочным руководством. Ее задача — дать читателю представление об эффективно используемых сейчас принципах, подходах и методах управления сложными объектами. А рецептуру, необходимую для конкретного применения изложенных подходов, можно найти в книгах, приведенных в литературных комментариях, сопровождающих каждую главу.

В основу книги положен преддипломный курс, читаемый автором вот уже более десяти лет на факультете автоматики и вычислительной техники Рижского политехнического института. Это обстоятельство определило синтетический и обобщающий характер изложения, кото-

рый опирается на представления, сложившиеся у студентов за все годы обучения. Это же позволило (и заставило) рассмотреть почти все разрозненные методы современной кибернетики с единых позиций управления сложным объектом.

Для понимания содержания книги достаточно знания математики в объеме вузовского курса: специальные вопросы оптимизации рассмотрены в приложении. Для беглого ознакомления с основными идеями и принципами управления сложными объектами можно прочитать лишь введение (для этого знание математики вовсе не обязательно). Если читатель выйдет за рамки этого введения, то цель автора будет достигнута.

С рукописью книги ознакомились ее рецензенты В. А. Абчук, И. А. Полетаев, чьи многочисленные критические замечания заставили автора значительно переработать книгу, Ю. И. Неймарк, добрые советы которого были учтены в окончательной редакции, А. Н. Складчиков, тщательность которого всегда помогает автору, а также В. П. и Ф. П. Тарасенко, советы которых позволили уточнить важные обстоятельства управления сложными объектами. Ценные замечания сделали по гл. 3 и 4 А. Н. Рубан, по гл. 5 — Ю. П. Адлер и А. М. Корилов, по гл. 6 — Ф. М. Кириллова. Всем им автор приносит глубокую благодарность.

Предисловие

...все простое — правда, а все сложное — неправда.

Г. СКОВОРОДА

Каждый из нас хоть раз в жизни выступал в роли руководителя, и всех нас волнуют вопросы управления и их принципы. Именно поэтому давайте сразу оговоримся, что здесь мы будем рассматривать управление объектами, поведение которых удастся описать формально, т. е. используя математический аппарат и современные электронные вычислительные машины (ЭВМ). И не будем путать такое управление с руководством, которое представляет собой управление пока неформализуемыми объектами (например, государством, коллективом, личностью и т. д.).

Управление — вещь сложная, тонкая и важная. Однако разговор об управлении хочется начать не со сложности, а ... с простоты. По глубокому убеждению автора, все научные принципы и идеи всегда и везде (в том числе и в области управления) простоты в своей основе. Но эта простота порой бывает обманчива.

Путь от сложного к простому прям и ясен: было сложно, разобрались — стало просто и ясно. Но есть и другой путь. Он связан, как правило, с применением формального подхода, т. е. с использованием математического аппарата. Не будем терять времени на выяснение того, зачем нужен формальный подход. Просто без него очень часто нельзя эффективно решить поставленную задачу, особенно в области управления. В любом случае формальный подход позволяет использовать ЭВМ. Формализм нужен, и без него не обойтись, особенно тогда, когда мы хотим переложить трудности решения задачи на ЭВМ. Это остро ощущается при решении задач управления сложными системами: все здесь настолько запутанно и сложно, что мысль о привлечении вычислительной машины для «распутывания»

этой сложности кажется спасительной. Но именно здесь и начинается усложнение и без того сложной задачи, т. е. ее формализация.

Специалисты (нематематики) не любят применять этот подход и обращаются к нему лишь в крайнем случае по многим причинам. Во-первых, в дебрях формализма чувствуют себя хорошо только математики, а специалисту-нематематику здесь трудно, так как он опирается на содержательный (физический) смысл задачи, ему же вместо этого предлагают всевозможные уравнения, алгоритм, преобразования и т. д. Но это еще не самое страшное. Не понимая толком всех математических премудростей, специалист, особенно если он не очень твердо понимает, чего же он хочет, перестает верить себе, и, как говорят, «математика торжествует над здравым смыслом». К сожалению, эта ситуация не придумана автором.

Именно поэтому так важно в процессе формализации задачи и ее решения формальными методами сохранить в неприкосновенности тот самый спасительный здравый смысл, который так ценится в неформальных подходах. А это по силам только специалисту, понимающему возможности формального подхода и трезво оценивающему его сильные и слабые стороны.

Нет формализма «на любой случай». Нас будет интересовать формализм, необходимый для управления. Здесь термин «управление» применяется в самом широком смысле: как совокупность действий, направленных на достижение поставленных целей. Под такое определение, как легко видеть, подпадают в принципе все разумные действия. Однако далеко не все их удастся формализовать. Мы в этой книге будем рассматривать лишь формальные стороны управления, позволяющие решать задачу управления сложным объектом с помощью ЭВМ. В строке остаются социальные, психологические и другие гуманитарные аспекты управления, которые оказывают влияние на некоторые коэффициенты уравнений или приводят к появлению некоторых очевидных ограничений.

В настоящее время разработаны многочисленные эффективные формальные методы синтеза управления, опирающиеся как на определенные математические модели типа идентификации, планирования экспериментов, математического программирования, так и на нефор-

мальные методы решения задач, реализуемые экспертами. Сочетание этих двух методов формального и неформального позволяет решать сложнейшие задачи управления и образует основу для управления сложными объектами. Именно это обстоятельство характеризует процессы управления сложными объектами и представляет собой основной аспект данной книги. Экспертный подход проявляется в большей мере на первых этапах управления, а формальный — на последующих.

В.1. Что такое управление?

Едва ли целесообразно в настоящее время вводить строгое определение понятия управления. Дело в том, что эта область знаний (теории и практики управления) со времени выхода книги Н. Винера о кибернетике настолько бурно развивается, что само понятие управления претерпевает неизбежные изменения, отражающие его новые аспекты как в теоретическом, так и прикладном плане.

Американского математика Норберта Винера (1894—1964) по праву считают создателем кибернетики. Он назвал ее «наукой об управлении и связи в животном и машине». Однако это определение уже не полно. Действительно, сейчас проблема связи вышла за рамки кибернетики. Шире понимается и область применения — она распространяется и на общество. Мало кого устроит управление вообще: нужно в определенном смысле наилучшее, *оптимальное* управление (ради достижения этой оптимальности и создается наука управления). Поэтому сейчас под кибернетикой понимают науку об оптимальном управлении сложными динамическими системами, подразумевая и машину, и животное, и общество, и любую другую сложную систему. Нас в этой книге будут интересовать в основном технические объекты управления, хотя описанные принципы могут быть применены для управления объектами более широкой природы.

Под *управлением* будем понимать процесс организации такого целенаправленного воздействия на объект, в результате которого объект переходит в требуемое (целевое) состояние. В этом определении, как и во всяком другом, одно понятие подменяется другими. Это, прежде всего, понятия «объект», «цель», «воздействие», «состояние». Хотя указанные понятия интуитивно очевидны и понимаются всеми примерно одинаково, они все же нуждаются в пояснении.

Объектом управления будем называть ту часть окружающего мира, состояние которой нас интересует и на которую мы можем воздействовать целенаправленно, т. е. управлять ею. Обозначим буквой X воздействие среды на объект (это вход объекта), а буквой Y — состояние объекта (это его выход). На рис. В.1.1. объект изображен в виде прямоугольника. Не будем пока конкретизировать эти обозначения (это будет сделано в последующих главах). Важно то, что об объекте и о X и Y мы можем говорить содержательно, т. е. описывать их на каком-то удобном нам языке (необязательно формальном).

Например, если объектом управления является автомобиль, то X определяет состояние его среды, т. е. дорожного покрытия, атмосферы, освещенности, а также



Рис. В.1.1. Взаимодействие объекта со средой

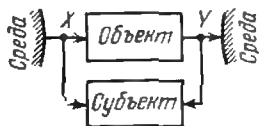


Рис. В.1.2. Схема взаимодействия субъекта со средой и объектом

положение его органов управления (шофер для автомобиля тоже среда); Y — состояние автомобиля, определяемое в основном скоростью движения, давлением масла, температурой и оборотами двигателя и т. п. Очевидно, что состояние объекта Y зависит от состояния его среды X , т. е. между ними имеется какая-то связь, характер и вид которой пока может быть и неизвестен. Это обстоятельство на рис. В.1.1 изображено графически, но его же можно представить в виде формулы

$$Y = F^0(X), \quad (\text{В.1.1})$$

где F^0 обозначает связь между входом и выходом объекта. Эту связь в математике для статических объектов называют функцией, а для динамических — оператором. Таким образом, в самом общем случае F^0 является неизвестным пока оператором, связывающим вход X и выход Y объекта. Этот оператор характеризует специфику объекта с точки зрения управления.

Говоря об управлении как о целенаправленном процессе, нельзя не обратить внимание на то, чьи цели реа-

лизуются в процессе управления, т. е. кому нужно это управление. Поэтому совершенно необходимо ввести в рассмотрение фигуру *субъекта*, который является источником целей, реализуемых управлением. Эти цели возникают у субъекта под давлением его собственных потребностей, связанных с жизнедеятельностью субъекта и его взаимодействием с внешней средой и объектом управления. (Заметим, что под субъектом совершенно не обязательно подразумевать конкретную личность; это может быть группа людей, объединенная по некоторому признаку, и даже все человечество, если рассматривается управление глобальными объектами, такими, как окружающая среда, космос и т. д.)

Появление при анализе управления субъекта с его потребностями совершенно необходимо, если мы хотим понять, как образуются цели управления.

Говоря о причинах управления, следует помнить известное правило криминалистов при расследовании преступления «Ищите, кому это выгодно!». Применяя это правило к управлению, мы приходим ... к нашему субъекту: только ему нужно и выгодно управление.

Так как фигура субъекта важна для управления, выделим его из среды, как показано на рис. В.1.2. Субъект находится в той же среде, что и объект, т. е. воспринимает состояние X среды. Одновременно, и это самое главное, на него влияет состояние Y объекта. Если состояние Y объекта удовлетворяет потребностям субъекта, взаимодействующим с этим объектом и использующим его для своих целей, то никакого управления ему не нужно. Если же это состояние не устраивает субъекта, то ему необходимо организовать такое воздействие на объект, которое переведет его в новое состояние, удовлетворяющее субъекта. Это воздействие и есть управление. Отсюда и следует один из «принципов» управления: оно происходит от неудовлетворенности субъекта ситуацией, сложившейся в объекте.

При полном благополучии субъекта, т. е. при удовлетворении объектом всех потребностей субъекта, управление не нужно. Однако удобно (хотя это и условно) считать, что субъект всегда формулирует свою цель, реализация которой в объекте приведет, по мнению субъекта, к удовлетворению его потребностей. Эта цель представляет собой набор требований, предъявляемых субъектом к состоянию объекта.

Обозначим цель субъекта Z^* (звездочкой будем обозначать все желаемое). Эта цель по сути дела определяет, каким должен быть объект с точки зрения субъекта. Проверить выполнение цели Z^* в объекте можно только по его состоянию Y , но для этого состояние Y объекта следует выразить на языке целей субъекта, т. е. выполнить преобразование

$$Z = \Psi(Y). \quad (\text{B.1.2})$$

В частном случае может оказаться, что $Z = Y$, т. е. субъект формулирует свои цели на языке состояний объекта. Например, пусть цель Z заключается в том, чтобы скорость автомобиля была равна 60 км/ч. Спидометр на приборном щитке показывает действительную скорость Y в тех же единицах (км/ч). В этом случае $\Psi(Y) = Y$, так как нет необходимости «переводить» показания спидометра на язык нашей цели: это уже сделано. Но если мы едем на иностранной машине, в которой скорость определяется в милях/ч, то $\Psi(Y) \neq Y$. Действительно, в этом случае $Z = 0,63 Y$, т. е. приходится фиксированное прибором состояние объекта Y [миль/ч] «переводить» на язык целей субъекта [км/ч].

Очевидно, что равенство

$$Z = Z^* \quad (\text{B.1.3})$$

свидетельствует о том, что состояние объекта удовлетворяет целям субъекта, т. е. цели субъекта выполнены. Если же

$$Z \neq Z^*, \quad (\text{B.1.4})$$

то цели субъекта не реализованы в этом объекте. Это последнее обстоятельство заставляет субъекта решить дилемму: либо смириться с Z и в результате терпеть определенный ущерб, связанный с недостижением своих целей Z^* , а следовательно, с неудовлетворением потребностей, либо создать систему управления, которая реализовала бы его цели Z^* в объекте, но при этом затратить определенные средства на ее создание и эксплуатацию. (Здесь под системой управления понимаются все мероприятия, связанные с выработкой и реализацией управления.)

На «смирение» идут в большинстве случаев. Этот принцип даже подтверждается народной мудростью. Говоря строго, терпение — тоже управление, но внутрен-

нее, направленное на сдерживание и ограничение потребностей субъекта. Это управление потребностями. В какой-то мере это даже разумно. Действительно, если можно терпеть, то острой надобности в управлении нет. К управлению обращаются как к крайней мере.

Дело в том, что за управление надо платить и часто дорогой ценой! Это прежде всего затраты на сбор информации об объекте, на синтез его модели, на выработку управления и реализацию его и т. д. и т. п. Именно поэтому к управлению обращаются в исключительных случаях. Однако такие исключительные случаи встречаются слишком часто (виной тому растущие потребности субъекта), что и заставляет разрабатывать общие методы управления, рассматриваемые в этой книге.

Если цели важные (т. е. потребности, их породившие, насущные), то субъект идет на создание системы управления. Для этого ему прежде всего необходимо определить, каким образом можно воздействовать на объект, т. е. определить каналы управления. Ими могут быть некоторые из входов X объекта, которые поддаются целенаправленному изменению. Но обычно этого бывает мало, и приходится создавать новые каналы управления объектом, которые до этого не существовали. Так, чтобы изменять направление движения телеги, пришлось создать руль, с помощью которого можно воздействовать на ориентацию колес. Это и есть новый канал управления.

Так или иначе в целом ряде случаев для реализации управления создаются специальные каналы управления. Обозначим буквой U это управление. Очевидно, что теперь состояние объекта управления зависит от двух факторов — состояния среды (X) и состояния управления (U):

$$Y = F^0(X, U), \quad (B.1.5)$$

где F^0 — по-прежнему оператор работы объекта, но теперь он учитывает еще и управление U .

Теперь уже можно говорить о создании *системы управления*, под которой будем понимать все необходимые алгоритмы обработки информации и средства их реализации, объединенные для достижения заданных целей управления в объекте. Заметим, что система управления далеко не всегда реализуется «в металле». Она может быть реализована в виде системы правил, договоров и обязательств (вообще алгоритмов), которые

реализуются в процессе управления. Так, систему управления образует любой оператор вместе с объектом, которым он управляет. Например, шофер за рулем автомашины образует с ней систему управления. В этом случае органы управления автомашины (руль, педаль газа, тормоз и т. д.) являются теми каналами управления U , с помощью которого шофер изменяет состояние автомашины Y в желаемом ему направлении Z^* . В данном случае «управляющее устройство» (шофер) и субъект, задающий цель Z^* , совпадают.

В.2. Система управления

Схема системы управления показана на рис. В.2.1. Здесь D_X и D_Y — датчики, с помощью которых измеряется состояние среды и объекта соответственно. Результаты измерений

$$X_d = D_X(X); Y_d = D_Y(Y), \quad (\text{В.2.1})$$

где D_X и D_Y — операторы этих датчиков, являются исходной информацией для управляющего устройства (УУ), которое на этой основе вырабатывает управляющее воздействие U .

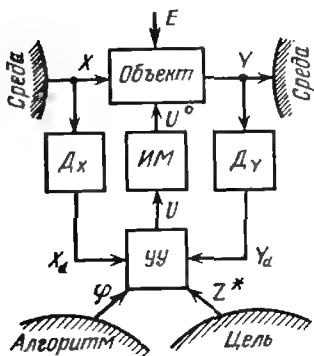


Рис. В.2.1. Схема системы управления

Эти формулы выражают очевидную связь между состоянием (среды или объекта) и информацией об этом состоянии. А это далеко не одно и то же. Разницу легко увидит тот, кто попытается ознакомиться с состоянием Черного моря по информации, содержащейся в фотографиях и рассказах. В этом случае X — действительное состояние Черного моря, а X_d — рассказы о нем.

Все, что нас окружает, мы ощущаем с помощью наших пяти органов чувств и действуем (управляем) на основе этих наших ощущений. Если управление доверить ЭВМ, то для этого ее следует снабдить «органами чувств» — указанной системой датчиков D_X и D_Y о состоянии среды (X) и объекта (Y).

Таким образом, в основе процесса управления лежит информация о сложившейся ситуации:

$$I = \langle X_d, Y_d \rangle, \quad (B.2.2)$$

хотя она всегда является неполной. Эта неполнота, прежде всего, связана с ограниченными возможностями всякой системы сбора информации I и необходимостью «платы» за эту информацию. А при ограниченных ресурсах, выделяемых обычно на управление, указанное обстоятельство приводит к постоянному дефициту информации о поведении среды и объекта.

Сказанное заставляет образовать еще один вход объекта — ненаблюдаемое возмущение E (см. рис. B.2.1), под которым подразумеваются все ненаблюдаемые внешние и внутренние факторы объекта, влияющие на его состояние Y , т. е. $Y = F^0(X, U, E)$, что уточняет (B.1.5). Относительно входа E могут выдвигаться лишь определенные предположения, но непосредственно этот вход не измеряется. Вход E образуют прежде всего неизмеряемые параметры среды (например, температура внешней среды при автоматическом управлении температурой камеры холодильника) и всякого рода случайные изменения характеристик самого объекта (например, дрейф его характеристик).

Таким образом, объект управления имеет три входа: наблюдаемый (X), управляемый, а следовательно, и наблюдаемый (U^0) и ненаблюдаемый (E). Заметим, что управляющее воздействие (команда, вырабатываемая управляющим устройством) U представляет собой информацию о том, в какое положение должны быть приведены управляемые входы объекта U^0 . «Отработка» этой команды осуществляется исполнительными механизмами (ИМ), которые и изменяют состояние управляемого входа U^0 объекта. Будем для простоты предполагать, что исполнительные механизмы мгновенно выполняют команду U и поэтому $U^0 = U$, т. е. можно не различать U^0 и U (случай $U^0 \neq U$ исследуется теорией автоматического управления, а точнее теорией следящих систем, хорошо и весьма полно разработанной).

Для целенаправленного функционирования управляющего устройства ему кроме информации (B.2.2) необходимо сообщить цель Z^* управления, т. е. к чему ему следует стремиться в процессе управления, и алгоритм управления, т. е. указание, как добиться этой цели.

Итак, введены очень важные понятия: *цель* и *алгоритм*. Если понятие цели интуитивно ясно — это то, к чему мы стремимся, то понятие алгоритма нуждается в пояснении. Под алгоритмом в данном случае мы понимаем четкое недвусмысленное правило, инструкцию, указание, что и как следует сделать, чтобы добиться заданной цели Z^* в сложившейся ситуации I .

Например, любая инструкция по применению лекарства, прилагаемая к нему, является алгоритмом, так как она указывает, в каких случаях (I) следует, а в каких не следует применять это лекарство. Целью (Z^*) здесь является выздоровление больного, а управление — дозировка лекарства (нулевая дозировка тоже является управлением). Но инструкция «иди туда, не знаю куда, принеси то, не знаю что» не может быть названа алгоритмом, хотя и имеет его структуру.

Теперь, располагая информацией о состояниях среды, объекта и цели, можно представить управление U как результат работы алгоритма:

$$U = \Phi(I, Z^*), \quad (B.2.3)$$

где Φ — алгоритм управления. Он представляет собой оператор $\Phi = \Phi(\cdot, \cdot)$, перерабатывающий информацию о среде, объекте и цели в управление U , реализация которого U^0 должна переводить объект в требуемое состояние Z^* .

Таким образом, основными факторами всякого управления, которые присущи любой системе управления объектом, являются:

- цель управления (Z^*);
- информация о состоянии объекта и среды (I);
- воздействие на объект, т. е. собственно управление (U);
- алгоритм управления (Φ).

Заметим, что слово «управление» обычно используют в двух смыслах. В широком смысле это процесс достижения поставленных целей (в этом смысле мы его применяли раньше), а в узком — всякое целенаправленное воздействие на объект (нажатие кнопки, поворот руля, выделение ресурса и т. д.).

Если исключить из перечня основных факторов управления хотя бы один, управление станет невозможным. Именно этим обстоятельством, например, во время военных действий пользуются диверсионные группы, на-

правляя свои удары против системы управления войсками противника. Объектом диверсии чаще бывают системы коммуникации, с помощью которых передается информация I и управляющее воздействие U , а также штабы, где реализуется алгоритм управления войсками.

Таким образом, управление в широком смысле определяется множеством целей $\{Z^*\}$, которые поступают в систему управления извне (фигурными скобками $\{a\}$ всюду мы будем обозначать множество элементов a). Эти цели ставит субъект, являющийся потребителем будущей системы управления объектом. Субъект выступает в качестве заказчика на создание системы управления. Созданием же этой системы управления занимается проектировщик.

Однако, прежде чем принять решение о создании системы управления, необходимо рассмотреть все этапы управления, независимо от того, с помощью каких материальных средств будут реализованы эти этапы. Такой алгоритмический анализ управления является основой для принятия решения о создании системы управления и степени ее автоматизации.

При этом анализе следует учитывать фактор сложности объекта управления. Дело в том, что управление простым объектом в настоящее время не представляет труда. Примером управления простым объектом является автоматический процесс поддержания температуры на заданном (целевом) уровне, который реализуется в любом холодильнике. Здесь алгоритм управления температурой крайне прост: включить охлаждение, если температура в камере выше заданной, и выключить, если — ниже или равна заданной. Вот и все!

Все трудности управления определяются сложностью объекта, поэтому следует начинать именно с объекта, а не с алгоритма управления им. Попробуем разобраться, что называть *простым*, а что *сложным* объектом управления без ссылки на управление. К сожалению, кратким определением здесь ограничиться не удастся. Поэтому придется посвятить этому вопросу целый параграф.

В.3. Что такое «сложный объект управления»?

Это понятие в настоящее время еще строго не определено. Возможно, что такое определение никогда и не появится, а если и появится, то тогда, когда в нем, по

сути дела, отпадет острая необходимость. Интуитивное представление о сложном объекте довольно точно соответствует тому понятию, которое используется в теории управления (оно-то и применяется в этой книге). Приведем некоторые черты (но не формальные признаки) сложной системы. Их много. Рассмотрим основные.

1. Отсутствие математического описания и необходимость в нем является обязательной чертой сложного объекта управления. Под математическим описанием подразумевается наличие алгоритма (правил, инструкции) вычисления состояния Y объекта по наблюдениям его входов — управляемого U и неуправляемого, но наблюдаемого X . С этой точки зрения современная баллистическая (или космическая) ракета не является сложной системой, так как ее движение описывается уравнениями классической механики (законами Ньютона), которые легко составить и решить, коль скоро известны конструктивные параметры ракеты и свойства среды. Поэтому поведение ракеты более определено, чем поведение муравья, которого по праву следует отнести к сложной системе, так как его поведение пока даже и не пытаются описать математически.

Простые объекты управления могут тоже не иметь математического описания. Так, для управления температурой в комнате совершенно не обязательно знать, какова математическая зависимость между температурой нагревателя и температурой в этой комнате. Однако если простыми объектами можно управлять без их математической модели, то сложными уже нельзя. Так, при необходимости очень точного поддержания заданной температуры без математической модели уже не обойдешься. Здесь требование точности делает простой объект управления сложным. Этот вывод позволяет сформулировать следующее утверждение: если хочешь управлять сложной системой с использованием формальных методов, то придется создать ее математическую модель. Это обстоятельство обычно охлаждает горячие головы, желающие управлять «всем и вся» с помощью вычислительной машины.

2. Стохастичность поведения сложных объектов управления является также очень важной чертой, характеризующей трудности процессов анализа и управления ими. Эта черта обусловлена не столько наличием каких-то специальных источников случайных помех в объ-

екте управления, сколько сложностью объекта и связанным с этим неизбежным обилием всякого рода второстепенных (с точки зрения целей управления, разумеется) процессов. Поэтому его поведение зачастую оказывается неожиданным для исследователя, причем эту неожиданность удобнее рассматривать как случайный фактор и трактовать как зашумленность, чем разбираться в механизме второстепенных процессов, протекающих в сложном объекте. Любой сложный объект содержит большое число такого рода неожиданностей, которые и являются свидетельством его сложности. Таковы биологические, социальные, технологические и многие другие объекты исследования и управления.

Здесь проявляется любопытная тенденция — называть случайным то, что второстепенно и несущественно для реализации целей управления в этом объекте. В действительности, если разобраться, эта «случайность» может оказаться вовсе и не случайной. Но в данном случае очень удобно считать случайным все то, что непонятно. Этот подход хорошо развит в настоящее время в виде статистической теории управления (заметим, что понятия «статистический» и «стохастический» эквивалентны).

3. «Нетерпимость» к управлению является, пожалуй, самой досадной чертой сложного объекта управления, затрудняющей управление им. Дело в том, что сложный объект существует и функционирует независимо от субъекта и его потребностей. Так, любой технологический процесс существует для производства продукции, а не для управления им. Управление имеет внешний характер по отношению к объекту. Естественно, что вследствие этого всякое управление нарушает «нормальное» функционирование объекта, т. е. изменяет его самостоятельное поведение и делает зависимым от субъекта. Особенно наглядно это обстоятельство проявляется в активных объектах, содержащих в себе людей или их коллективы (например, когда объектами управления являются технологический участок, цех или завод). Здесь трудно рассчитывать на то, что собственные цели такого сложного объекта совпадут с целями управления. Скорее они будут в чем-то противоречить друг другу. Это и вызывает негативную реакцию сложного объекта на управление, если цель управления не согласована с его собственной целью.

2* * объект имеет свою цель,
* управление объектом имеет
свою цель.

4. Нестационарность сложного объекта управления естественно вытекает из его сложности. Эта черта проявляется в дрейфе характеристик объекта, в «уплывании» его параметров, т. е. в эволюции объекта во времени. Чем сложнее объект, тем рельефнее проявляется эта его черта, что создает серьезные трудности при синтезе модели такого сложного объекта и управлении им. Чем сложнее объект, тем быстрее он меняется. В обыденной жизни с этим процессом мы сталкиваемся, наблюдая амортизацию (старение) окружающих нас вещей и предметов. Причем эта амортизация для сложных вещей идет, как правило, быстрее, чем для простых. Так, телевизор или автомобиль «стареют» быстрее, чем молоток. Следствием нестационарности является следующая существенная черта сложного объекта.

5. Невоспроизводимость экспериментов. Проявляется эта черта в различной реакции объекта на одну и ту же ситуацию или управление в различные моменты времени. Сложный объект все время как бы перестает быть самим собой, т. е. постоянно меняется. Этого нельзя не учитывать при синтезе модели объекта управления, для чего вводится ее коррекция.

Все эти обстоятельства приводят к тому, что цель управления таким объектом в полной мере никогда не достигается. Действительно, для синтеза управления U требуется время, за которое объект изменяется непредвиденным образом, в результате это управление уже наверняка не приведет к желаемому результату Z^* .

Эффективным способом борьбы с перечисленными свойствами сложного объекта управления является экстраполяция поведения системы, т. е. выяснение направления ее эволюции. В этом случае управление U производится с упреждением, с учетом замеченного изменения объекта. Другой способ состоит в сокращении цикла управления, чтобы за время синтеза управления объект сильно не изменился.

Список «досадных» черт сложного объекта управления можно было бы продолжить. Однако следует помнить, что это лишь черты, присущие сложному объекту, но ни в коей мере не его формальные признаки. Отсутствие одной или даже нескольких из указанных черт вовсе не обязательно делает объект простым. Эти черты имеют неформальный характер, но тем не менее позволяют характеризовать сложный объект управления.

В.4. Этапы управления сложным объектом

Управление сложным объектом состоит из следующих этапов (рис. В.4.1).

1. *Формулировка целей управления $\{Z^*\}$* . На этом этапе определяются цели (точнее, множество целей), которые должны быть реализованы в процессе управления. Слово «цель» здесь используется в смысле «модели будущего», т. е. некоторого предполагаемого состояния объекта, которое желательно субъекту (потребителю)

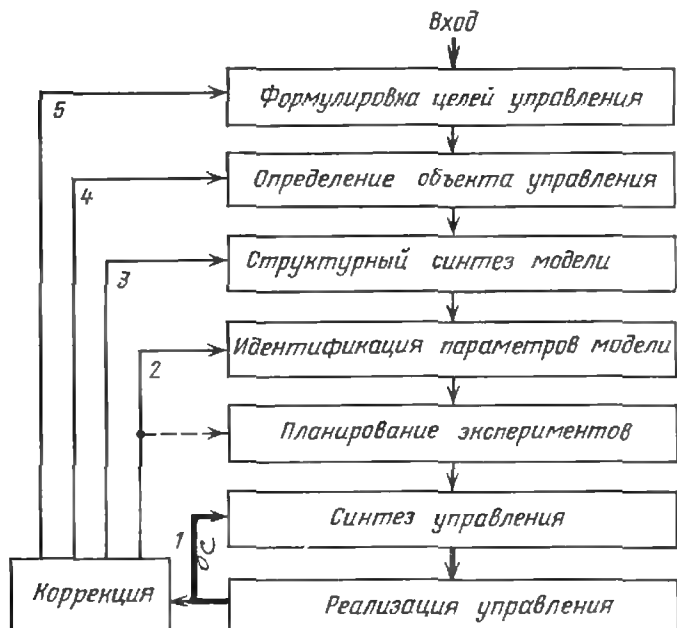


Рис. В.4.1. Этапы управления сложным объектом.

и которое в определенном смысле «неестественно», т. е. не реализуется естественным образом в природе без вмешательства извне, т. е. без управления. Проще говоря, на этом этапе выясняется вопрос о том, что же хочется субъекту и стоит ли этого хотеть. Если стоит, то не выполнится ли это желание само собой. И если само собой это желание не осуществляется, то оно становится целью управления.

Так, поддерживать температуру t в помещении на определенном необходимом уровне t^* независимо от состояния погоды можно только с помощью управления. Это управление может быть автоматическим (с помощью кондиционера) или неавтоматическим (ручным), но в любом случае целью такого управления является Z^* : $t=t^*=\text{const}$, причем константа t^* может изменяться в зависимости от обстоятельств (потребностей субъекта).

Задание множества целей $\{Z^*\}$ в данном случае сводится к определению диапазона температур, внутри которого может когда-либо при каких-то неизвестных обстоятельствах располагаться целевая (т. е. необходимая субъекту) температура t^* . Автоматическое управление в данном случае связано с аппаратной реализацией алгоритма: автоматически включать нагреватель при $t < t^*$ и выключать его при $t > t^*$. Так работает терморегулятор. Однако этот же алгоритм можно реализовать и в ручном неавтоматическом варианте.

2. *Определение объекта управления.* Этот этап связан с выделением той части среды потребителя, состояние которой его интересует в связи с реализацией сформулированных им целей. Это и есть объект управления. Например, выбранный диапазон целевых температур определяет объект управления, т. е. ту часть пространства, состояние которой необходимо контролировать (измерять температуру) и на которую следует воздействовать (подогревать или охлаждать в различных местах), чтобы выполнить заданные цели управления $\{Z^*\}$.

Часто объект управления ясен и определен с самого начала (автомобиль, радиоприемник, завод и т. д.). Но в ряде случаев, когда он связан со средой многочисленными и сильными связями, определить его нелегко. Попробуйте определить сферу вашего управления (т. е. объект) при поддержании чистоты в квартире. Квартира? Но грязь на лестничной клетке заставляет жаловаться коменданту. А это — управление, целью которого является все та же чистота в вашей квартире. Вот объект и расширился: им стала и лестница. Отсутствие асфальтовых дорожек у дома заставит вас обратиться в райисполком все с той же целью поддержания чистоты в квартире, т. е. вашим объектом управления становится двор дома, и т. д.

Последующие три этапа управления связаны с созданием модели объекта управления, которая нужна для

синтеза управления. Только с помощью модели объекта можно построить управление U , переводящее сложный объект в требуемое (целевое) состояние Z^* . Без модели процесс управления можно реализовать лишь методом проб и ошибок, что неприемлемо при управлении сложным объектом, так как требует слишком больших затрат времени и заставляет «дергать» объект ошибочными действиями (шагами).

Под моделью объекта управления будем понимать зависимость F , связывающую состояние Y объекта с его входами — неуправляемым X и управляемым U :

$$Y = F(X, U). \quad (\text{B.4.1})$$

В отличие от оператора объекта F^0 в (B.1.5), который неизвестен, модель F представляет собой высказывание относительно связи между входами X и U модели и ее выходом на любом удобном языке. В общем случае модель F определяется некоторым алгоритмом (правилом, инструкцией), который указывает, как, располагая информацией о входах X и U , определить выход Y , не обращаясь к реальному объекту. Процесс синтеза модели объекта обычно происходит в три этапа.

3. *Структурный синтез модели.* На этом этапе определяется вид зависимости F без учета значений ее параметров. Произведем условно следующую операцию: «расщепим» модель F на ее структуру St и параметры $c_1, \dots, c_k$, т. е. представим модель в виде пары:

$$F = \langle St, C \rangle, \quad (\text{B.4.2})$$

где $C = (c_1, \dots, c_k)$ — вектор параметров модели (векторы всюду будем обозначать полужирными буквами). На этапе структурного синтеза определяется лишь структура St модели, а конкретными значениями параметров C не интересуются.

Под структурой в общем случае принято понимать вид элементов, из которых состоит объект, и отношений между элементами. Описывать структуру удобнее всего графом, вершины которого — элементы объекта, а дуги — отношения, в которые вступают элементы объекта. Так, завод как объект управления состоит из цехов, отделов, служб и т. д. (это его элементы), связанных между собой отношениями подчиненности, материальных и информационных потоков и т. д.

Структура объекта управления может быть различной в зависимости от целей управления. Проиллюстри-

руем эту мысль на следующем (не очень серьезном) примере. Один и тот же объект — завод, имеет различную структуру в зависимости от целей управления. Так, с точки зрения целей министерства завод является преобразователем выделяемых фондов (денежных и материальных) в готовую продукцию. Работники завода рассматривают его как преобразователь их труда в зарплату (мы намеренно упростили схему). Медицинские учреждения относятся к тому же заводу как к источнику всякого рода производственных травм и профессиональных заболеваний, а санитарная служба считает, что завод производит отходы, загрязняющие среду.

Существует много разнообразных структур моделей объекта управления. Линейность, статичность, детерминированность, дискретность и т. д. являются структурными категориями. Например, линейная статическая непрерывная детерминированная структура объекта однозначно определяет следующий вид для F :

$$Y = F(X, U) = c_0 + c_1 X + c_2 U,$$

где конкретные значения параметров c_0, c_1, c_2 пока не важны, важен лишь вид зависимости F , т. е. линейность структуры St .

Таким образом, на этапе структурного синтеза определяется лишь вид и характер модели F , а ее параметры C определяются на последующих двух этапах управления.

4. *Идентификация параметров модели* объекта связана с определением численных значений параметров $C = (c_1, \dots, c_k)$ в режиме нормального функционирования, т. е. без организации специальных управляющих воздействий*) на объект. Исходной информацией для идентификации являются структура St и наблюдения за поведением входа $X(t)$ и выхода $Y(t)$ объекта при его взаимодействии со средой. Таким образом, пара

$$I(t) = \langle X(t), Y(t) \rangle \quad (B.4.3)$$

является в общем случае основным источником информации при идентификации.

*) Заметим, что иногда под идентификацией подразумевают процесс определения параметров модели безотносительно к тому, в каком режиме работает объект. Поэтому различают пассивную и активную идентификацию. В этой книге рассматривается пассивная идентификация, а планирование эксперимента является активной идентификацией (см. следующий пункт).

Искусство идентификации — это искусство пассивного наблюдения. Таково было основное кредо древней науки: уметь наблюдать не вмешиваясь. На это есть свои основания, так как всякое вмешательство искажает естественный ход наблюдаемого явления. Более того, результаты эксперимента некоторым образом предопределяются его целями. Очень часто в эксперименте мы получаем не то, что есть, а, грубо говоря, то, что хочется, и связано это не с недобросовестностью экспериментатора, а с психологией процесса познания. Именно поэтому пассивное наблюдение остается и сейчас эффективным инструментом познания окружающего нас мира.

Необходимость пассивного наблюдения связана еще и с тем, что часто сложные объекты не допускают экспериментов с ними. По-видимому, эти соображения породили нелепое понятие «пассивного эксперимента». Эксперимент не может быть пассивным — он всегда активен! Речь идет просто о пассивном наблюдении.

Как легко заметить, информация (В.4.3) позволяет определить зависимость состояния объекта лишь от входа X : $Y = F(X)$, которая отличается от искомой модели (В.4.1). Для выяснения зависимости выхода Y объекта от управляемых входов U необходимо преднамеренно их изменять, т. е. «поэкспериментировать» с объектом. Однако, как сказано выше, сложный объект «не любит» экспериментов, нарушающих режим его нормального функционирования. Поэтому эксперимент, которого нельзя избежать, следует проводить, минимально возмущая объект, но так, чтобы получить при этом максимальную информацию о влиянии варьируемых параметров на выход объекта. Этому служит следующий этап — планирование экспериментов.

5. Планирование экспериментов. На этом этапе главным является синтез плана эксперимента, позволяющего с максимальной эффективностью определить параметры модели объекта управления.

Для статического объекта этот план $\mathcal{U}$ представляет собой набор состояний управляемого входа объекта:

$$\tilde{U} = \langle U_1, \dots, U_N \rangle, \quad (\text{В.4.4})$$

а для динамического — план-функцию

$$\tilde{U} = U(t), \quad 0 \leq t \leq T, \quad (\text{В.4.5})$$

т. е. программу изменения во времени управляемого входа объекта. Эксперимент на объекте позволяет определить реакцию объекта на это воздействие. Для статического объекта эта реакция образует ряд

$$Y_1, \dots, Y_N, \quad (\text{B.4.6})$$

где $Y_i = F^0(U_i)$, $i = 1, \dots, N$, — реакция объекта в i -м эксперименте, а для динамического $Y(t) = F^0(U(t))$.
Теперь N пар I_i :

$$I_i = \langle U_i, Y_i \rangle, \quad i = 1, \dots, N, \quad (\text{B.4.7})$$

для статического объекта и пара I :

$$I = \langle U(t), Y(t) \rangle \quad (\text{B.4.8})$$

для динамического объекта являются исходной информацией для определения параметров C модели F :

$$Y = F(U). \quad (\text{B.4.9})$$

Как видно, после проведения эксперимента задача определения параметров модели сведена к задаче идентификации (см. предыдущий этап).

Для получения модели вида (B.4.1) необходимо рассматривать уже тройки наблюдений I_i :

$$I_i = \langle X_i, U_i, Y_i \rangle, \quad i = 1, \dots, N, \quad (\text{B.4.10})$$

или

$$I = \langle X(t), U(t), Y(t) \rangle, \quad 0 \leq t \leq T, \quad (\text{B.4.11})$$

т. е. в процессе планирования экспериментов с объектом необходимо учитывать и состояние среды X .

Следует отметить, что в процессе планирования экспериментов с объектом у экспериментатора могут измениться представления о структуре модели (например, первоначальная гипотеза о линейности модели будет заменена гипотезой о нелинейности определенного вида). Это обстоятельство заставляет снова обращаться к структурному синтезу, точнее, вводить коррекцию структуры модели. Сказанное несколько «размывает» понятие планирования эксперимента, распространяя его и на процессы выбора структуры модели и на ее коррекцию.

Понятие планирования эксперимента чрезвычайно емкое и насыщенное. Планировать эксперименты можно не только для выяснения параметров модели, о чем мы

уже сказали, но и для выбора одной из предполагаемых структур и т. д. Если, например, в качестве объекта управления выступает человек, то можно планировать эксперименты по выяснению некоторых его специфических свойств, представляющих интерес для управления (скажем, для выяснения профессиональных склонностей при профотборе). Такие эксперименты называются психологическими тестами. Испытуемому при этом предлагают решить ряд задач (тестов) и по ответам судят о его склонностях, т. е. образуют модель испытуемого. Составление подобных тестов производится методами планирования экспериментов, с помощью которых выясняется «принадлежность» испытуемого к той или иной профессиональной структуре.

Таким образом, на этапе планирования экспериментов определяются параметры модели, которые нельзя определить на этапе идентификации, и корректируется структура модели. Но цель обоих этапов — окончательное создание модели F объекта управления F^0 . На этом задачу синтеза модели, которой были посвящены три последних этапа (3, 4 и 5), можно считать выполненной. Полученная модель является исходной для процесса синтеза управления.

6. *Синтез управления.* Этот этап связан с принятием решения о том, каково должно быть управление U , чтобы достигнуть заданной цели Z^* управления объектом. Это решение опирается на имеющуюся модель F объекта, заданную цель Z^* , полученную информацию о состоянии X среды и выделенный ресурс управления R , который чаще всего представляет собой ограничения, накладываемые на управление U в связи со спецификой объекта и возможностями системы управления (временными, энергетическими, материальными и т. д.).

Естественно синтезированное управление U считать оптимальным, т. е. $U=U^*$, так как оно должно удовлетворить поставленной цели Z^* управления. В общем случае оно представляет собой программу изменения управляемых параметров во времени, т. е.

$$U^*=U^*(t). \quad (B.4.12)$$

Эту программу предстоит реализовать на следующем этапе. Примером программы оптимального управления, которое каждый из нас испытал на себе, является план медицинского лечения, прописываемого врачом: «Ну-с,

батенька, будем делать так . . .», и далее следует подробный план U^* лечебных мероприятий, который учитывает: цель Z^* (побыстрее сделать больного «практически» здоровым при минимальном ущербе, наисосимом лечением, т. е. минимальном риске осложнений и т. п.), состояние X среды (время года, жилищные условия больного, наличие заботливых родственников и т. д.), ресурс R больного (возможность достать дефицитные лекарства) и поликлиники (наличие необходимых приборов и оборудования для диагностики и лечения), модель F объекта, т. е. представление врача о больном в данный момент с медицинской точки зрения (т. е. диагноз, анализы, переносимость больным определенных лекарств, дисциплинированность, психическое состояние и т. д.).

7. Реализация управления связана с реализацией программы оптимального управления (В.4.12), полученной на предыдущем этапе. Этот процесс не вызывает затруднений, если сведения о состояниях среды X , объекта Y и его модели F были достоверны. Однако к моменту реализации все могло измениться. Для учета такого рода изменений требуется соответствующая коррекция управления U в процессе его реализации. Исходной информацией для такой коррекции является новое состояние среды к моменту реализации управления или прогноз этого состояния. Если объект активный, т. е. содержит людей, цели которых могут не совпадать с целями управления, то в процессе реализации управления следует учитывать и это обстоятельство. В этом случае, очевидно, не удастся в полной мере достичь поставленных целей, но следует действовать так, чтобы отклонение от них было минимальным.

Поэтому план оптимального управления $U^*(t)$ и его реализация часто отличаются так же, как благие намерения в жизни от их осуществления. Реализовать управление часто бывает труднее, чем синтезировать. Например, планы всех оптимальных диет для похудения нарушаются потому, что они требуют реализации, т. е. каких-то ограничений в еде, что просто невыносимо для толстяков. Здесь неучет этого обстоятельства и делает нереализуемым такое управление.

Реализовав управление и получив новое состояние объекта, можно обнаружить, достигнута поставленная цель или нет. Скорее всего цель управления не будет

достигнута (напомним, что мы управляем сложным объектом, который благодаря своей зашумленности и нестационарности не позволяет ограничиться однократным актом управления). Поэтому приходится возвращаться к одному из предыдущих этапов. Даже в самом лучшем случае, когда поставленная цель оказывается достигнутой, необходимость обращения к предыдущему этапу вызывается изменением состояния среды X или сменой цели управления Z^* . Таким образом, при самом благоприятном стечении обстоятельств обращаются к этапу синтеза управления для определения нового управления, которое отражает новую ситуацию, сложившуюся в среде, и новую цель. Так функционирует стандартный контур управления (см. стрелку 1 на рис. В.4.1).

8. *Коррекция.* Этот этап обусловлен спецификой сложного объекта управления и заключается в возвращении к одному из предыдущих этапов управления (см. стрелки 2—5 на рис. В.4.1). Дело в том, что все решения, принимаемые на предыдущих этапах, приближенные, опираются на старую информацию и отражают состояние объекта лишь в прошедшие моменты времени.

Коррекция может затрагивать различные этапы. Простейшая коррекция связана с подстройкой параметров C модели (см. стрелку 2 на рис. В.4.1). Такого рода коррекцию обычно называют адаптацией модели. Иногда может оказаться, что оптимальное управление $U^*(t)$ не обеспечивает необходимой вариабельности (разнообразия) управляемого входа объекта для эффективной последующей коррекции параметров C модели. Тогда приходится обращаться к планированию экспериментов с объектом в процессе управления им путем добавления специальных тестовых сигналов к управлению (см. штриховую стрелку на рис. В.4.1). Такое управление называют дуальным, т. е. двойственным, так как его задачей является не только достижение целей управления, но и коррекция параметров модели объекта.

Однако коррекции параметров C модели может оказаться недостаточно, если изменилась структура объекта. Поэтому время от времени необходима коррекция структуры модели St , т. е. приведение ее в соответствие с изменяющейся структурой объекта (см. стрелку 3 на рис. В.4.1).

Далее коррекция может коснуться самого объекта, точнее, границы раздела объекта и среды. Это бывает

необходимо при значительном изменении (эволюции) объекта и окружающей его среды. В этом случае коррекция «узаконит» новые взаимоотношения нового объекта со средой (см. стрелку 4 на рис. В.4.1).

И последнее. Созданная система управления по ряду причин (значительное изменение свойств среды и объекта) может не реализовать все исходное множество целей управления $\{Z^*\}$, которые породили эту систему, или реализовать их недостаточно эффективно. Тогда возникает необходимость коррекции целей управления, которая заключается в определении нового множества целей, достигаемых с помощью этой системы управления. В результате этой коррекции потребитель узнает, какие же цели данная система управления действительно может реализовать в объекте (см. стрелку 5 на рис. В.4.1).

Реализация указанных мер и образует адаптивную систему управления, которая приспособляется к изменяющимся свойствам среды, объекта и потребностям ее потребителя. Любопытно, что очень часто потребителя удовлетворяет созданная система управления, несмотря на то, что она не реализует всех его исходных целей. Предоставленная ему возможность достижения каких-то новых целей (ранее ненужных ему) деформирует его потребности в осуществлении легкодостижимых целей. Так, часто мы едим не потому, что голодны, а потому, что так сложились обстоятельства, а «аппетит приходит во время еды». Легкость в достижении целей питания порождает именно эти цели. Особенно ярко этот механизм проявляется в капиталистическом обществе, где с помощью мощной рекламы предложение порождает спрос, т. е. потребность.

Заметим, что чем больше номер стрелки на рис. В.4.1, тем реже срабатывает эта обратная связь. Благодаря функционированию описанной системы коррекции система управления все время приводится в соответствие с изменяющимися средой и объектом управления. В результате система управления как бы эволюционирует вместе с объектом и средой, но так, чтобы все время хорошо выполнялись цели управления. Если же эта эволюция объекта управления заходит настолько далеко, что исходное множество целей перестает выполняться, то коррекция целей обеспечивает эволюцию целей управления.

Очевидно, что не все описанные выше восемь этапов управления всегда присутствуют при синтезе системы управления. В ряде случаев некоторых может и не быть. Например, когда объект управления уже выделен из среды (самолет, космический корабль и т. д.) или нет необходимости в этапе планирования эксперимента, так как модель объекта настолько проста и «прозрачна», что все ее параметры можно определить без специально организованного эксперимента. Может отсутствовать этап формулировки целей управления: настолько они очевидны и естественны. Природа не столь «злонамеренна», и при работе реальных систем управления редко приходится реализовать все этапы управления, как правило, некоторые этапы просто выпадают в силу специфики целей и объекта управления. Так, при управлении автомашиной не возникает проблемы выбора объекта управления, а при лечении больного цель всегда одна и та же — вылечить и не повредить. Но в любом случае при управлении сложным объектом должен существовать этап коррекции модели объекта управления. Необходимость этого этапа вызвана сложностью объекта управления. Только для простых объектов он может отсутствовать. Однако будем готовы к самому худшему и рассмотрим все восемь этапов, замкнутые в железное кольцо системы управления!

В.5. Как сделать коктейль? (Несерьезная иллюстрация серьезных этапов управления)

Проиллюстрируем рассмотренные в предыдущем параграфе этапы управления сложным объектом на шуточном примере решения задачи приготовления коктейля.

Итак, вам необходимо составить рецепт коктейля, который, с одной стороны, должен быть вкусным и достаточно крепким, а с другой — не слишком дорогим. Предстоит принять решение о том, что и в каком количестве покупать, чтобы получить то, что требуется. А что требуется? Какие конкретные цели должны быть достигнуты при этом?

Целевыми критериями всякого коктейля, как известно, являются его вкусовые качества, крепость и стоимость единицы объема. Пусть z_1 — вкусовые качества коктейля, z_2 — его крепость (например, в градусах),

z_3 — стоимость одного литра смеси (например, в рублях), u_i — относительное содержание i -го компонента коктейля, где, например, $i=1$ — фруктовый сок, $i=2$ — сухое вино, $i=3$ — коньяк и т. д. Всего имеется n компонентов ($i=1, \dots, n$), так что $\sum_{i=1}^n u_i = 1$ и $u_i \geq 0$.

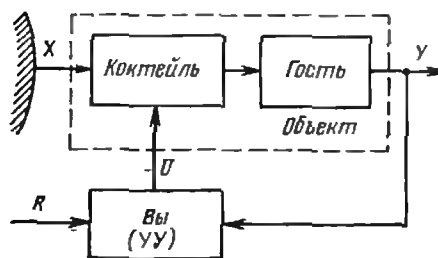


Рис. В.5.1. Схема «системы управления» для составления коктейля

Вектор $U = (u_1, \dots, u_n)$ определяет состав коктейля. От этого вектора каким-то определенным, но пока неизвестным образом зависят все три критерия:

$$z_1 = f_1(U), \quad z_2 = f_2(U), \quad z_3 = f_3(U),$$

где f_1 , f_2 и f_3 — пока неизвестные целевые функции.

Сформулируем задачу приготовления коктейля как задачу управления. Объектом управления в данном случае является сам коктейль и гость, который его оценивает (см. рис. В.5.1), неуправляемым входом X — исходные компоненты коктейля, а управляемым U — их относительное содержание. Выходом этого объекта являются свойства коктейля, оцененные вашим гостем, $Y = (z_1, z_2, z_3)$, а «управляющим устройством» (УУ) — бармен (это Вы) (рис. В.5.1).

Прежде всего сформулируем цели этого управления (этап 1). Они, как уже сказано, зависят от потребностей субъекта, т. е. потребителей коктейля — ваших гостей.

Если гость — лакомка, то основное внимание он уделяет вкусовым качествам коктейля, и естественно критерий z_1 (вкусовые качества коктейля) считать экстремальным критерием, который (точнее, целевую функцию $f_1(U)$) следует максимизировать в процессе управления. Критерий z_2 (крепость) следует ограничить снизу:

$z_2 \geq a_2$, где a_2 — минимальная крепость, которая устроит ваших гостей, а критерий z_3 (стоимость) нужно ограничить сверху: $z_3 \leq a_3$, где a_3 — максимальная стоимость одного литра коктейля, определяемая вашим бюджетом. Как легко заметить, этим определяется цель лакомки:

$$Z^* : \{z_1 \rightarrow \max; z_2 \geq a_2; z_3 \leq a_3\}.$$

Это означает, что потребности лакомки будут удовлетворены, если максимизировать вкусовые качества коктейля, а его крепость и стоимость ограничить числами a_2 и a_3 .

Легко определить цель любителей крепких напитков:

$$Z^* : \{z_1 \geq a_1; z_2 \rightarrow \max; z_3 \leq a_3\},$$

т. е. максимизировать крепость коктейля, ограничив при этом его минимальные вкусовые качества числом a_1 и максимальную стоимость одного литра числом a_3 .

Аналогично определяется цель ограниченного в средствах хозяина, минимизирующего стоимость коктейля:

$$Z^* : \{z_1 \geq a_1; z_2 \geq a_2; z_3 \rightarrow \min\}.$$

Как видно, для задания любой из указанных целей необходимо задать два из трех чисел: a_1 , a_2 , a_3 . Таким образом, для лакомки предстоит решить задачу вида

$$f_1(U) \rightarrow \max_{U \in \Omega_1}, \quad \Omega_1 : \begin{cases} f_2(U) \geq a_2, \\ f_3(U) \leq a_3, \end{cases} \quad (B.5.1)$$

т. е. следует ($\rightarrow$) максимизировать ($\max$) функцию $f_1(U)$, определяющую вкусовые качества коктейля, при определенных ограничениях, накладываемых на функции $f_2(U)$ и $f_3(U)$, определяющие зависимость крепости и стоимости коктейля от его состава U (буквой Ω_1 обозначено множество $\{U\}$, удовлетворяющее указанным ограничениям по крепости и стоимости).

Здесь впервые введена символика оптимизации, которая широко будет использоваться далее (общая постановка задачи оптимизации и эффективные способы ее решения даны в приложении).

Если субъекта больше волнует крепость коктейля, то предстоит решить уже другую задачу:

$$f_2(U) \rightarrow \max_{U \in \Omega_2}, \quad \Omega_2 : \begin{cases} f_1'(U) \geq a_1, \\ f_3(U) \leq a_3, \end{cases}$$

где следует ($\rightarrow$) с помощью состава U максимизировать ($\max$) крепость коктейля, сохранив его вкусовые качества и стоимость в оговоренных пределах (a_1 и a_3).

И, наконец, для субъекта, сильно ограниченного в средствах, естественно сформулировать задачу в таком виде:

$$f_1(U) \rightarrow \min_{U \in Q_3}, \quad \Omega_3: \begin{cases} f_1(U) \geq a_1; \\ f_3(U) \leq a_3. \end{cases}$$

т. е. варьируя состав U коктейля, следует ($\rightarrow$) минимизировать ($\min$) его стоимость, выполнив при этом определенные требования Ω_3 к вкусовым качествам и крепости коктейля.

Далее в соответствии с § В.4 и схемой рис. В.4.1 следует этап 2 — выделение объекта управления из среды. Но для данной задачи объект был определен ранее — это сам коктейль (гость здесь играет роль датчика).

На следующем, третьем, этапе — структурном синтезе модели объекта, необходимо определить характер управляемых и неуправляемых воздействий на объект и структуру модели, т. е. вид зависимости свойств коктейля от выбранных спиртных напитков и соков, а также от их относительного содержания.

Таким образом, в процессе структурного синтеза модели объекта надо определить вид функций $f_j(U)$ ($j = 1, 2, 3$). Прежде всего необходимо определить структуру управления, т. е. вид вектора $U = (u_i, \dots, u_n)$. Иными словами, следует содержательно описать каждую переменную, т. е. какой именно сок или вино «скрывается» под индексом i в переменной u_i , и задать размерность n вектора составом U . Далее следует определить характер зависимости интересующих нас функций от аргумента U .

Крепость и стоимость коктейля, как известно, строго линейно зависят от концентрации его компонентов:

$$f_1(U) = \sum_{i=1}^n c_{i2} u_i, \quad f_3(U) = \sum_{i=1}^n c_{i3} u_i,$$

здесь c_{i2} — относительная крепость i -го компонента. Например, сухому вину ($i=2$), имеющему крепость 10° , приписывается относительная крепость $c_{22}=0,1$; c_{i3} — стоимость одного литра i -го компонента. Чтобы опреде-

лить эти параметры, достаточно взглянуть на этикетки бутылок, где приведены крепость c_{i2} (в градусах) и стоимость c_{i3} (в рублях за объем, равный емкости бутылки).

А как быть с функцией вкусовых качеств $f_1(U)$? Как зависит вкус коктейля от соотношения его компонентов? Легко заметить, что она будет для каждого субъекта индивидуальной. А так как коктейль общий и время на приготовление ограничено, т. е. нет времени исследовать вкус каждого потребителя коктейля, то функцию f_1 следует задать достаточно простой. Простейшей является линейная функция

$$f_1(U) = \sum_{i=1}^n c_{i1} u_i,$$

где c_{i1} — «вкусовые» коэффициенты качества i -го компонента. Конечно, линейная функция лишь приближенно отражает действительную зависимость вкуса от содержания компонентов. Но возможно, что удачным выбором коэффициентов c_{i1} ($i=1, \dots, n$) удастся хорошо аппроксимировать эту функцию. На этом этапе структурный синтез модели объекта управления можно считать законченным.

Теперь нужно определить числовые значения «вкусовых» коэффициентов c_{i1} ($i=1, \dots, n$). Прежде всего, никаких объективных данных здесь не существует, да и не должно быть, так как давно известно, что «на вкус и цвет товарищей нет». Для оценки значений этих коэффициентов следует воспользоваться этапом идентификации, в процессе которого оцениваются коэффициенты c_{i3} ($i=1, \dots, n$) на основе информации, полученной из наблюдений за вашим гостем в аналогичных ситуациях. Если же таких данных нет, то следует приступить к этапу планирования эксперимента. Он может быть произведен различным образом. Проще всего спросить у гостя, что он больше предпочитает из того, что есть в вашем баре. Для этого потребителю будущего коктейля следует задать ряд таких вопросов, ответы на которые несли бы информацию, необходимую для определения искомых коэффициентов. Такой опрос является экспериментом с объектом, трудоемкость и сложность которого определяются числом вопросов и их трудностью. Если же в вашем баре есть соки и вина, которых

не знает ваш гость, то эксперимент следует расширить до их дегустации. Проведя необходимые эксперименты с будущими потребителями коктейля (как это делать, мы обсудим ниже в гл. 3), можно определить искомые значения «вкусовых» коэффициентов. На этом заканчивается синтез модели объекта.

Далее следует перейти к этапу принятия решения, точнее, синтезу управления, т. е. определению оптимального состава U^* . Для этого необходимо решить задачу (В.5.1), где функции f_i ($i=1, 2, 3$) уже известны (надемся, что вы выберете цель лакомки).

Рассмотрим к примеру случай трех компонентов ($n=3$): $i=1$ — фруктовый сок, $i=2$ — сухое вино, $i=3$ — коньяк. Получаем для них исходные данные:

j	c_{ij}	$i=1$ (сок)	$i=2$ (вино)	$i=3$ (коньяк)	Единица измерения	Источник информации
1	c_{11}	$c_{11}=1$	$c_{21}=0,75$	$c_{31}=0,5$	Вкус Относительная крепость Стоимость 1 л (руб.)	Экспертиза Этикетка „
2	c_{12}	$c_{12}=0$	$c_{22}=0,1$	$c_{32}=0,4$		
3	c_{13}	$c_{13}=1$	$c_{23}=3$	$c_{33}=16$		

Для определенности дальнейших рассуждений конкретизируем цель. Пусть $a_2=0,1$ — минимально допустимая крепость смеси, а $a_3=3,5$ — максимально допустимая стоимость 1 л (это константы цели лакомки). Теперь задачу синтеза оптимального состава коктейля можно записать так:

$$u_1 + 0,75u_2 + 0,5u_3 \rightarrow \max_{u_1, u_2, u_3 \in Q_1},$$

$$Q_1: \begin{cases} 0,1u_2 + 0,4u_3 \geq 0,1, \\ u_1 + 3u_2 + 16u_3 \leq 3,5, \\ u_1 + u_2 + u_3 = 1, \\ u_i \geq 0 \quad (i=1, 2, 3), \end{cases}$$

где в ограничения Q_1 дополнительно введены очевидные условия, налагаемые на u_i :

$$\sum_{i=1}^n u_i = 1 \text{ и } u_i \geq 0.$$

Исключая третье условие в Ω_1 ($u_3 = 1 - u_1 - u_2$), получаем после очевидных преобразований задачу в виде

$$0,5u_1 + 0,25u_2 + 0,25 \rightarrow \max_{u_1, u_2 \in \Omega'},$$

$$\Omega'_1: \begin{cases} 4u_1 + 3u_2 \leq 3, \\ 15u_1 + 13u_2 \geq 12,5, \\ u_1 + u_2 \leq 1, \\ u_1 \geq 0, u_2 \geq 0. \end{cases}$$

Область Ω'_1 показана на рис. В.5.2 (она не заштрихована). Как видно, эта зона представляет собой треугольник ABC в левом верхнем углу. Теперь на этом треугольнике надо найти точку, максимизирующую линейную функцию

$$z_1 = 0,5 u_1 + 0,25 u_2 + 0,5,$$

выражающую вкусовые качества коктейля. Очевидно, что максимальное значение линейная функция принимает на границе зоны Ω'_1 , а точнее, в одной из вершин треугольника ABC . Вычислим значения z_1 в этих вершинах:

$$z_1(A) = 0,75;$$

$$z_1(B) = 0,5 \cdot 0,21 + 0,25 \cdot 0,75 + 0,5 = 0,78;$$

$$z_1(C) = 0,25 \cdot 0,94 + 0,5 = 0,74.$$

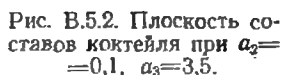
Следовательно, точка B и есть решение поставленной задачи. Таким образом, оптимальный состав первого коктейля таков:

$$u^*_1 = 0,21; u^*_2 = 0,72; u^*_3 = 0,07,$$

где u_3 вычислено с помощью выражения $u_3 = 1 - u_1 - u_2$. Вкусовые качества такого коктейля определяются значением $z^*_1 = 0,78$. Полученный рецепт и следует считать первым оптимальным решением. На этом заканчивается этап синтеза управления.

Пропустим этап реализации этого управления — едва ли он вызовет у Вас затруднения, хотя возможно, что отсутствие необходимых вин в ближайшем магазине (среде) и заставит Вас пересмотреть структуру модели, т. е. выбрать другие компоненты коктейля. Это и будет той самой коррекцией управления, вызванной изменением состояния среды к моменту реализации управле-

Перейдем к следующему этапу — коррекции. Реакция Y объекта (ваших гостей) на реализованное управление (коктейль) определяет справедливость (адекватность) модели. Едва ли Вы ожидали единодушного одобрения своего рецепта. Очевидно, что разноречивой в оценках должен быть. Однако средняя оценка при правильной модели должна совпадать с полученной в расчете, т. е. $z^*_1 = 0,78$. Если средняя экспертная оценка значительно отличается от этой (даже в лучшую сторону), то модель нуждается в коррекции, т. е. изменении параметров c_{i1} ($i = 1, \dots, n$). Например, если экспертная оценка кок-


$$c'_{i1} = (0,6/0,78) c_{i1} = 0,77 c_{i1}, \quad i = 1, \dots, n,$$

Если подобного рода коррекция не улучшает управление, то следует скорректировать структуру модели, например, используя нелинейную функцию вкусовых качеств коктейля или выбирая другие компоненты и т. д.

Так или иначе, но информация, полученная после реализации управления, является исходной для коррекции модели управляемого объекта. А если этот объект, как в нашем примере, является сложной системой с изменяющимися во времени свойствами за счет дрейфа ее характеристик, то коррекция модели неизбежна даже в гипотетическом случае точного синтеза модели в начальный момент.

Коррекция может затронуть и объект управления, если вы обратитесь на завод с просьбой выпускать новые сорта соков или вин. И, наконец, можно корректировать цели, изменяя параметры a_1 , a_2 и a_3 и переходя к другому минимизируемому критерию (вплоть до кофе или огуречного рассола).

В заключение отметим, что описанный пример был придуман автором только как забавная иллюстрация этапов управления сложной системой. Если же читатель попробует воспользоваться изложенным подходом для приготовления настоящего коктейля, автор снимает с себя всякую ответственность за последствия.

ЛИТЕРАТУРНЫЕ КОММЕНТАРИИ К ВВЕДЕНИЮ

Первые книги по кибернетике, написанные Н. Винером почти тридцать лет назад, до сих пор не потеряли своей значимости. Будучи классиком, Н. Винер умел простым языком излагать очень тонкие и глубокие аспекты новой науки. Поэтому читателю будет интересно прочитать отдельные главы из этих книг (см. [1, 2]).

О кибернетике написано много доступных книг. Среди них особое место занимают книга У. Эшби [3], в которой простота изложения удивительно сочетается с глубиной идей, и книга Ст. Бира [4], которому удалось, не применяя никакого математического аппарата, поставить и решить ряд интереснейших задач управления сложными объектами типа современного промышленного предприятия.

Научно-популярные издания по кибернетике обычно обсуждают различные аспекты этой науки [5—7]. Отметим [5] — первую книгу по кибернетике на русском языке.

Описанные в § В.1.4 восемь этапов управления сложным объектом являются конкретным руководством к действию при желании управлять таким объектом. Однако это желание может быть удовлетворено лишь в случае, если удастся формализовать объект и процесс управления. Если же такая формализация затруднительна или невозможна, то следует обращаться к так называемому системному подходу (или анализу), который представляет собой по сути дела некий полуформализованный здравый смысл. По этому поводу написано много книг и руководств. Отметим лишь основные.

Этапы системного подхода (например, изложенного В. М. Глушковым [8]) во многом совпадают с приведенными в § В.1.4 этапами управления сложным объектом. Основные моменты подхода очень

ясно изложены в книге [9], где сопоставлены и различные его модификации [10—13]. Интересные обобщения системного подхода имеются в книгах [14—16].

Следует отметить, что системный подход в управлении сложными объектами обычно связан с руководством этими объектами, т. е. с неформальными методами управления. Это очень хорошо показано в работе [15]. Именно поэтому системный подход и не нашел места в данной книге, которая посвящена формальным методам управления, а не руководству (см. предисловие).

Таким образом, с точки зрения системного подхода (или анализа) предложенную в книге восьмиступенчатую процедуру управления сложными объектами можно рассматривать как некий прикладной системный анализ, подразумевая под этим свод формальных приемов (сочетаемых с неформальными источниками информации — экспертами), позволяющих конструктивно решать задачу управления сложными объектами.

СПИСОК ЛИТЕРАТУРЫ

1. Винер Н. Кибернетика: Пер. с англ. — М.: Сов. радио, 1968, с. 5—97, 182—256, 285—313.
2. Винер Н. Кибернетика и общество: Пер. с англ. — М.: ИЛ, 1958.
3. Росс Эшби У. Введение в кибернетику: Пер. с англ. — М.: ИЛ, 1959.
4. Ст. Бир. Кибернетика и управление производством: Пер. с англ. — М.: Физматгиз, 1963.
5. Полетаев И. А. Сигнал. — М.: Сов. радио, 1958.
6. Растринин Л. А. Этот случайный, случайный, случайный мир. — М.: Молодая гвардия, 1974.
7. Растринин Л. А., Граве П. С. Кибернетика как она есть. — М.: Молодая гвардия, 1975, 208 с.
8. Глушков В. М. Введение в АСУ. — Киев: Техника, 1974, с. 146—163.
9. Черняк Ю. И. Системный анализ в управлении экономикой. — М.: Экономика, 1975, с. 41—60.
10. Оптнер С. Л. Системный анализ для решения деловых и промышленных проблем. — М.: Сов. радио, 1969.
11. Федоренко Н. П. О методах социально-экономического прогнозирования. — В кн.: Методология прогнозирования экономического развития СССР. — М.: Экономика, 1971.
12. Янг С. Системное управление организацией: Пер. с англ. — М.: Сов. радио, 1972.
13. Клиланд Д., Кинг В. Системный анализ и целевое управление: Пер. с англ. — М.: Сов. радио, 1974, с. 25—37.
14. Основы системного подхода и их приложение к разработке территориальных автоматизированных систем управления. — Томск: Изд-во Томского университета, 1976, с. 3—145.
15. Джонсон Р., Каст Ф., Розенцвейг Д. Системы и руководство: Пер. с англ. — М.: Сов. радио, 1971.
16. Системное проектирование АСУ хозяйством области./ Под ред. Ф. П. Перегудова. — М.: Статистика, 1977.

ФОРМУЛИРОВКА ЦЕЛЕЙ УПРАВЛЕНИЯ

В основе всякого управления лежит информация о том, чего собственно добивается, что хочет субъект от объекта, т. е. каковы его цели $\{Z^*\}$ (точнее, множество целей). Именно эта информация определяет объект и систему управления им. Процесс формулировки целей предшествует всей процедуре управления и поэтому в значительной мере формирует ее.

1.1. Парадокс цели

Между целью и объектом, в котором реализуется эта цель, вообще говоря, существует двусторонняя связь: цель влияет на выбор объекта, а объект, в свою очередь, определяет характер цели. Эта взаимная обусловленность цели и объекта связана с тем, что субъект не может формулировать свою цель, не имея хотя бы какого-то предварительного (априорного) представления об объекте, в котором эта цель должна быть реализована. Дело в том, что проверка выполнения (или невыполнения) заданной цели осуществляется путем анализа состояния Y объекта. Поэтому для проверки реализуемости цели просто необходимо знать, что наблюдать (измерять) в объекте. Это означает, что в процессе формулирования целей следует иметь довольно четкие представления об объекте будущего управления.

Таким образом, существует своеобразный парадокс: цель определяется объектом, а объект — целью. Этот парадокс разрешается довольно просто. Субъект, формулируя цель, всегда имеет какие-то представления об объекте. Эти представления могут быть очень приближенными, но всегда отражают некоторые его свойства, достаточные для эффективной формулировки целей управления. Именно поэтому формулировка цели с той или иной степенью точности почти всегда возможна.

Объект управления целиком и полностью определяется целью управления. Следует, однако, помнить, что далеко не всякий объект и не всегда полностью удовлетворяет целям управления, т. е. далеко не все сформулированные цели достижимы при имеющихся ресурсах,

выделяемых на управление. Действительно, всегда можно сформулировать такие нереальные цели, которые при ограниченном ресурсе ни на каком объекте не могут быть реализованы. Это обстоятельство обычно заставляет корректировать цели управления, что, в свою очередь, неизбежно изменяет объект управления и т. д. Именно поэтому цель не возникает отдельно от объекта. По-видимому, субъект обдумывает различные варианты цели и объекта одновременно, они связаны друг с другом и активно взаимодействуют. Однако с методологической точки зрения удобно разделить эти процессы, т. е. синтез цели рассмотреть независимо от процесса выделения объекта из среды, хотя в действительности они идут параллельно.

Таким образом, отмеченный выше парадокс цели удобно разрешить, исключив обратное влияние объекта на цель. Это влияние можно учесть на следующем этапе, после того как определен объект и изучены его возможности с точки зрения достижения поставленной цели. Если окажется, что не существует объекта, в необходимой мере удовлетворяющего поставленной цели, то нужно скорректировать цель, так как она в данном случае недостижима.

Теперь рассмотрим процесс образования целей и исследуем его специфику. Здесь уместно вспомнить, что цели, о которых идет речь в этой главе, выдвигаются субъектом. Поэтому естественно разобраться, что представляет собой субъект, чьи цели мы предполагаем реализовать с помощью синтезируемой системы управления. Для этого следует рассмотреть модель субъекта.

1.2. Модель субъекта

Прежде всего отметим, что субъект является активной стороной и отражает в процессе своего взаимодействия со средой не только состояние этой среды, но и свои собственные интересы и потребности, связанные с жизнедеятельностью. Заметим, что при управлении большими сложными объектами в качестве субъекта может выступать не только один человек, но и большая группа людей. Поэтому под субъектом мы, вообще говоря, будем понимать активную систему, чьи интересы должны быть отражены в системе управления (ради них она и создается).

Пусть у субъекта имеется k различных потребностей, каждую из которых мы будем характеризовать числом a_i — степенью ее выраженности, насущности, актуальности. Очевидно, что потребности изменяются во времени в зависимости от состояния субъекта и среды, в которой он находится. Таким образом, в каждый момент времени субъект характеризуется набором (вектором) своих потребностей:

$$A = (a_1, \dots, a_i, \dots, a_k). \quad (1.2.1)$$

Естественно считать, что

$$a_i \geq 0, i = 1, \dots, k, \quad (1.2.2)$$

т. е. актуальность потребности не может принимать отрицательных значений и при отсутствии i -й потребности $a_i = 0$. Например, в простейшем случае можно рассмотреть лишь три примитивные потребности: жажда, голод и секс ($k=3$). Такой субъект, очевидно, будет формулировать лишь «животные» цели, что и определит соответственно характер его поведения.

Забегим вперед и предположим, что субъект умеет достигать поставленные цели, т. е. он располагает системой реализации этих целей (это и есть система управления). Тогда его поведение будет сводиться к формулировке цели управления и к достижению этой цели, т. е. к осуществлению равенства

$$Z = Z^* \quad (1.2.3)$$

с помощью управления U . Напомним, что Z — состояние объекта, описанное на языке целей $\{Z^*\}$. «Перевод» с одного языка на другой осуществляется функцией Ψ , рассмотренной выше (В.1.2): $Z = \Psi(Y)$, где $\Psi(\cdot)$ — заданная функция, определенная на состояниях Y объекта. Не будем рассматривать, как субъект добивается реализации целей управления, т. е. как он определяет U . Достаточно знать, что он каким-то образом умеет это делать, т. е. уже располагает эффективной системой управления, которая «отрабатывает» заданные ей цели Z^* .

Взаимоотношения субъекта с этой системой управления (СУ) показаны на рис. 1.2.1, где штриховой линией выделена СУ, реализующая заданную ей цель Z^* . Субъект здесь выступает как датчик целей, отрабаты-

ваемых СУ. При этом потребности субъекта удовлетворяются с помощью канала Y .

Задача субъекта, таким образом, сводится к тому, чтобы эффективно сформулировать цель управления. Для этого он прежде всего должен уметь отличать «хорошие» цели от «плохих». Говоря языком математики, субъект должен иметь специальную функцию меры качества цели $\mu(Z^*)$, т. е. своеобразную функцию неком-

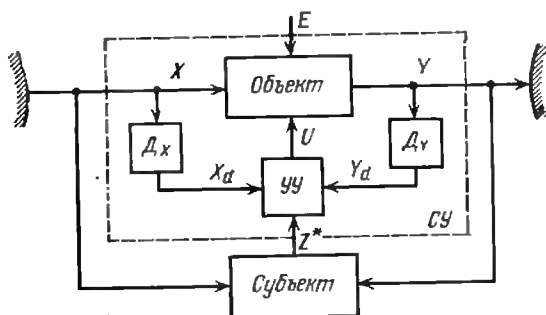


Рис. 1.2.1. Схема взаимодействия субъекта с системой управления

фортности, которая определена на множестве всех возможных целей $\{Z^*\}$. Эта функция позволяет сравнивать две цели Z^*_1 и Z^*_2 следующим образом. Если

$$\mu(Z^*_1) < \mu(Z^*_2), \quad (1.2.4)$$

то цель Z^*_1 предпочтительнее («лучше») цели Z^*_2 , что выражается так:

$$Z^*_1 > Z^*_2, \quad (1.2.5)$$

где $>$ — знак предпочтения (читается «более предпочтительно, чем»). При $\mu(Z^*_1) = \mu(Z^*_2)$ цели эквивалентны, т. е. $Z^*_1 \sim Z^*_2$. Заметим, что цели могут быть эквивалентны для субъекта, но никак не совпадать.

Таким образом, выбор оптимальной цели сводится к минимизации функции $\mu(Z^*)$, т. е. к решению задачи

$$\mu(Z^*) \rightarrow \min_{Z^* \in \{Z^*\}} \Rightarrow Z^{**}. \quad (1.2.6)$$

Это выражение обозначает: следует ($\rightarrow$) минимизировать ($\min$) функцию $\mu(\cdot)$, варьируя ее аргумент Z^*

в пределах заданного множества целей $\{Z^*\}$, которое состоит из допустимых целей. Результатом решения этой задачи является (это обозначается знаком $\Rightarrow$) Z^{**} — наилучшая в данных обстоятельствах цель.

Приведем примеры образования меры $\mu(\cdot)$. Ее можно определить, например, как взвешенную сумму:

$$\mu = \sum_{i=1}^k b_i \alpha_i, \quad (1.2.7)$$

где $b_i > 0$ ($i=1, \dots, k$) — «веса» потребностей, характеризующие значимость соответствующей потребности для жизнедеятельности субъекта. Эти величины определяются с помощью экспертных оценок, т. е. путем соответствующим образом организованного опроса специалистов, хорошо знающих потребности субъекта (см. § 2.4).

Можно предложить и другую зависимость:

$$\mu = \max_{i=1, \dots, k} b_i \alpha_i, \quad (1.2.8)$$

которая характеризуется наиболее актуальной важнейшей потребностью.

Как легко заметить, величина μ определяет некомфортность субъекта, которую он, естественно, желает минимизировать с помощью выбора соответствующей цели Z^* , реализация которой наибольшим образом снижает эту некомфортность.

Реализация выбранной цели должна изменять потребности субъекта в сторону их уменьшения (для «хороших» целей):

$$A_1 \xrightarrow{Z^*} A_2, \quad (1.2.9)$$

где A_1 — исходные, а A_2 — полученные в результате реализации цели Z^* потребности (1.2.1). Естественно потребовать, чтобы при этом A_2 минимизировало функцию (1.2.7) или (1.2.8) при $b_i = \text{const}$. Но для этого необходимо установить связь (1.2.9) в виде зависимости

$$A_2 = \psi(A_1, Z^*, S), \quad (1.2.10)$$

где S — ситуация, сложившаяся в окружающей среде к моменту выбора цели Z^* , в простейшем случае $S = \langle X, Y \rangle$, т. е. это состояние среды, окружающей субъект.

Если бы эта зависимость была известна, то, подставляя ее в (1.2.7) или в (1.2.8), мы получили бы искомую

функцию $\mu(Z^*)$, минимизация которой однозначно определила бы оптимальную цель субъекта. Однако зависимость (1.2.10) пока может реализоваться лишь человеком в виде рассуждений следующего рода: «Мои потребности A_1 . Если я в ситуации S поставлю цель Z^* и достигну ее, то мои потребности станут равными A_2 . Если A_2 меня удовлетворяет, то я выбираю эту цель. В противном случае я пробую другую цель и т. д.». Мы пока можем только предполагать, каким образом субъект определяет состояние своих потребностей A_2 по исходным состояниям ситуации S потребности A_1 и цели Z^* . В этом ему помогают имеющиеся модели среды, собственной персоны и опыт применения этих моделей.

Таким образом, описанный механизм образования оптимальных целей управления может быть положен в основу создания формального «датчика целей». Однако для этого мы должны располагать двумя функциями: функцией дискомфорта $\mu(A)$ в виде (1.2.7) или (1.2.8) или в каком-то другом виде и функцией изменения потребностей $\psi(\cdot, \cdot, \cdot)$ (1.2.10), которая даст возможность определить потребности, возникающие в результате реализации цели Z^* . Пока с этой задачей в полной мере справляется только человек. Однако это не означает, что процедуру формулировки цели следует пустить «на самотек». Существуют вполне определенные способы выражения целей, и, коль скоро цель осознана, она должна быть выражена с помощью вполне определенных терминов, образующих язык формулировки целей Z^* . Итак, рассмотрим этот язык, т. е. определим, каким образом можно формально задавать цель Z^* .

1.3. Пространства ситуаций и целей

Субъект в процессе общения с окружающей его средой фиксирует свое внимание прежде всего на тех ее параметрах (свойствах) $s_1, \dots, s_k$, которые влияют на его потребности, т. е. на величины a_i ($i=1, \dots, k$), определяющие состояние дискомфорта субъекта. По значениям этих параметров субъект может судить о состоянии среды. Но, будучи активным, субъект, естественно, желает изменить эту среду так, чтобы уменьшить свою дискомфортность в ней, т. е. управлять ею. Это и заставляет его искать в среде такие параметры, изменить которые он в состоянии, причем так, чтобы это изменение понизило его дискомфортность в этой среде

(напомним, что объекта еще нет, его еще предстоит выделить из среды, из той ее части, которую захочет изменять субъект). Естественно считать, что субъект, образуя свои цели, учитывает только те параметры, с помощью которых он может улучшить свое состояние в этой среде. Так, например, желая увеличить скорость автомобиля, водитель обращает внимание только на два фактора его среды: положение педали газа и показания спидометра. Тормоз, сцепление и другие факторы в этот момент его не волнуют.

Параметры, которые определяют потребности субъекта, но не могут быть им изменены, вообще говоря, косвенно влияют на его поведение при целеобразовании. Здесь вступает в «игру» механизм эмоций, образуя то или иное настроение (например, раздражение), что не может оказать влияния на процесс образования цели. Так, неудовлетворенное желание обогнать при недостаточной мощности двигателя не может не вызвать нервозности водителя и в конечном счете повлияет на его поведение. Однако механизмы такого рода здесь рассматриваться не будут.

Таким образом, субъект воспринимает окружающую среду как некую ситуацию, которая описывается конечным или бесконечным набором параметров:

$$S = (s_1, \dots, s_l), \quad (1.3.1)$$

каждый из которых «волнует» субъекта, так как влияет на его дискомфортность, и может быть им изменен, т. е. воспринимаемая им ситуация всегда управляема:

$$S(U) = [s_1(U), \dots, s_l(U)], \quad (1.3.2)$$

где $U = (u_1, \dots, u_q)$ — управление субъекта, т. е. $u_1, \dots, u_q$ — управляемые параметры среды.

Например, для водителя контролируемые параметрами ситуации являются: s_1 — скорость автомашины, s_2 — положение (разворот) колес и т. д., а управляемыми: u_1 — положение педали газа, u_2 — положение рулевого колеса и т. д.

Теперь введем понятие пространства ситуаций $\{S\}$. Оно образуется указанными параметрами s_i ($i=1, \dots, l$) (1.3.1). Каждая точка этого пространства определяет какую-то конкретную ситуацию, сложившуюся вокруг субъекта (рис. 1.3.1). В это пространство $\{S\}$ отражаются окружающие субъекта среда и различные объекты.

Естественный дрейф ситуации, вызванный изменением (эволюцией) среды, связанным, возможно, с действиями других субъектов, приводит к смещению точки S вдоль какой-то траектории (см. линию на рис. 1.3.1).

Таким образом, $s_1, \dots, s_l$ — это параметры среды субъекта, которые влияют на него, воспринимаются им и могут быть им изменены посредством определенных воздействий U на среду.

Однако свои цели субъект формулирует не в терминах состояния среды S . Для этого ему удобнее оперировать иными свойственными ему понятиями, которые

представляются в виде переменных z_i ($i=1, \dots, p$) (назовем их целевыми). Пусть эти переменные описываются вектором

$$Z = (z_1, \dots, z_p), \quad (1.3.3)$$

где каждая целевая переменная z_i однозначно определяется ситуацией S , т. е.

$$z_i = \psi_i(S), \quad i=1, \dots, p, \quad (1.3.4)$$

а функции $\psi_i(\cdot)$ определяют связь состояния среды S и целевого параметра z_i . В каждый конкретный момент z_i является числом, характеризующим значение этого целевого параметра (например, температуры, длины, стоимости и т. д. или определяет наличие (1) или отсутствие (0) определенного свойства среды (например, наличие каких-то важных субъекту факторов). В векторной форме эта связь выражается в виде

$$Z = \Psi(S), \quad (1.3.5)$$

где

$$\Psi(\cdot) = (\psi_1(\cdot), \dots, \psi_p(\cdot)) \quad (1.3.6)$$

— определенная заданная вектор-функция, связывающая состояние среды и целевые переменные.

Преобразование Ψ информации к виду Z необходимо еще и потому, что субъект обычно формулирует свои цели в терминах и понятиях, связанных, но не тождественных измеряемому. В частном случае может оказаться, что $Z=Y$ ($p=l$), но это бывает редко. Например, при управлении температурным режимом объекта достаточно измерять температуру, т. е. $z=y=t$. Однако для

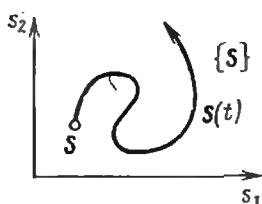


Рис. 1.3.1. Траектория дрейфа (эволюции) $S(t)$ ситуации в пространстве ситуаций

создания комфортных условий оператору необходимо измерять не только температуру, но и влажность, в то время как цель формулируется в виде ограничения на их определенную комбинацию.

Рассмотрим p -мерное пространство целей $\{Z\}$, которое образуется точками (1.3.3). Это пространство удобно субъекту тем, что по поводу каждой его координаты он может высказать свое требование (цель), выполнение которого, по мнению субъекта, приведет к удовлетворению какой-то или нескольких из его потребностей A .

Свою цель субъект формулирует в виде условного вектора цели

$$Z^* = (z^*_1, \dots, z^*_p), \quad (1.3.7)$$

где z^*_i — уже не числа, а требования к состоянию S среды, выраженные целевой переменной $z_i = \psi_i(S)$. Эти цели-требования могут иметь различный характер, но форма их представления должна быть унифицирована. Их необходимо свести к одной из следующих форм:

1) z^*_i — приравнять: $z_i = a_i$; это требование означает, что i -я целевая переменная $z_i = \psi_i(S)$ должна быть равна заданной величине a_i ;

2) z^*_j — ограничить: $z_j \geq b_j$, что выражает требование, налагаемое на j -ю целевую переменную, она не должна быть меньше заданного порога b_j ;

3) z^*_l — минимизировать: $z_l \rightarrow \min$, т. е. l -я целевая переменная должна быть минимальна.

Следовательно, для задания целей необходимо определить прежде всего их структуру, т. е. принадлежность к одной из этих трех форм, а затем задать числа a_i и b_j ($i=1, \dots, k_1$; $j=1, \dots, k_2$), где k_1 и k_2 — количества целей первого и второго рода. Если какие-либо цели субъекта не могут быть сведены к одной из этих форм, то нельзя говорить о формальном создании системы управления для достижения этих целей. Однако практика показывает, что к такого рода требованиям («приравнять», «ограничить» и «минимизировать») можно свести почти все ситуации, которые встречаются субъекту и для управления которыми создается система управления.

Заметим, что вначале цели субъекта имеют аморфный, неопределенный характер. В лучшем случае их удастся описать на языке предпочтений, т. е. указать наиболее предпочтительные ситуации, к которым стре-

мится субъект. Анализ этих ситуаций и дает возможность построить целевой вектор (1.3.7).

Таким образом, процесс формулировки целей субъекта связан, во-первых, с определением вектор-функции (1.3.6) и, во-вторых, с указанием требований, налагаемых на каждую составляющую этого вектора.

Пример. Измеряется температура в m точках печи: $s_1, \dots, s_m$. Необходимо так управлять температурным режимом этой печи (например, изменяя во времени подачу топлива), чтобы перепад температур в соседних точках не превышал заданного значения Δ .

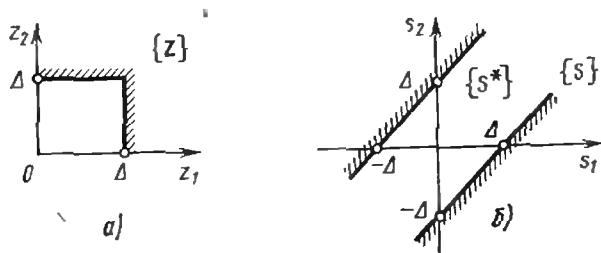


Рис. 1.3.2. Взаимодействие пространства целей (а) и ситуаций (б) в примере управления температурным режимом печи

В этой задаче m -мерное пространство ситуаций образуется показаниями датчиков $s_1, \dots, s_m$, а целевое пространство $\{Z\}$ имеет размерность $m-1$ и определяется функциями:

$$z_i = \psi_i(S) = |s_{i+1} - s_i|, \quad i = 1, \dots, m-1.$$

Цель Z^* в этом случае представляется неравенствами вида (рис. 1.3.2, а)

$$Z^*: z_i \leq \Delta, \quad i = 1, \dots, m-1 = k_2.$$

Рассмотрим, как отражается цель Z^* в пространство ситуаций $\{S\}$. Для этого достаточно рассмотреть область S^* , определяемую системой всех целевых требований:

$$S^*: \begin{cases} \psi_i(S) = a_i, & i = 1, \dots, k_1, \\ \psi_j(S) = b_j, & j = 1, \dots, k_2, \\ \psi_l(S) \rightarrow \min, & l = 1, \dots, k_3, \end{cases} \quad (1.3.8)$$

где k_3 — число экстремальных целевых требований. Целевая точка или область S^* , удовлетворяющая всем требованиям одновременно, и является тем состоянием среды, которого добивается субъект. Однако удастся ему это или нет, зависит от его возможностей воздействовать на эту среду, т. е. от вида зависимости (1.3.2)

и от ресурсов R управления, которыми располагает субъект. Ресурсные ограничения приводят к тому, что управление U субъекта ограничено

$$U \in \{U\}_R, \quad (1.3.9)$$

где $\{U\}_R$ — множество управлений, ограниченное ресурсами R , который определяется энергетическими, материальными и другими возможностями субъекта.

Рассмотрим целевую область $\{S^*\}$ пространства ситуаций для рассмотренного выше примера. Она определяется неравенствами

$$\{S^*\}: |s_{i+1} - s_i| \leq \Delta, \quad i=1, \dots, m-1.$$

На рис. 1.3.2,б показано пространство ситуаций $\{S\}$ с этой целевой областью $\{S^*\}$. Как видно, задание целевого требования Z^* удобнее делать в целевом пространстве $\{Z\}$. Задать целевую область $\{S^*\}$ в $\{S\}$, минуя стадию Z^* , в общем случае затруднительно.

Теперь рассмотрим взаимодействие целевой области $\{S^*\}$ и траектории изменения среды $S(t)$ под действием внешних факторов (рис. 1.3.3), т. е. дрейф ситуации. Если эта траектория проходит через целевую область, никакого управления субъекту не нужно. Ему достаточно дожидаться, когда внешние обстоятельства приведут к тому, что $S(t) \subset \{S^*\}$. Однако рассчитывать на подобное маловероятное совпадение неразумно, хотя оно в принципе и возможно. Именно поэтому субъект предпочитает управлять ситуацией, т. е. он целенаправленно воздействует на среду и изменяет ее:

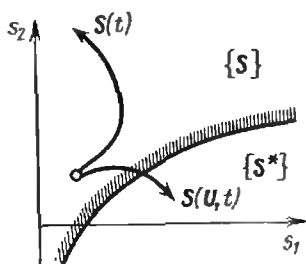


Рис. 1.3.3. Взаимодействие траекторий дрейфа $S(t)$ и управления $S(u, t)$ с целевой зоной $\{S^*\}$

$$S = S(U, t), \quad (1.3.10)$$

так, чтобы

$$S(U, t) \subset \{S^*\}, \quad (1.3.11)$$

т. е. чтобы достигались его цели. В этих выражениях время t обозначает влияние дрейфа на свойства среды.

Таким образом, управление U необходимо субъекту для того, чтобы добиться достижения цели управления и скомпенсировать дрейф ситуации, который, как правило, нарушает целевое условие (1.3.11).

Именно поэтому всякое управление следует рассматривать с двух точек зрения: как средство достижения поставленных целей и как средство компенсации неблагоприятных изменений в среде, препятствующих этому.

Заметим в заключение, что под параметрами ситуации S , сложившейся в среде, здесь и в дальнейшем подразумеваются измеряемые параметры собственно среды X и объекта Y , которые могут изменяться субъектом с помощью управления U , т. е. $Y = F^0(X, U)$. Таким образом,

$$S = \langle X, F^0(X, U) \rangle. \quad (1.3.12)$$

Однако такая дифференциация S возникает лишь после выделения объекта из среды, которое рассмотрено в следующей главе.

ЛИТЕРАТУРНЫЙ КОММЕНТАРИЙ К ГЛАВЕ I

Примат цели в задачах управления часто оспаривается на основе того, что в биологических системах управления, поражающих своей сложностью, целесообразностью и надежностью, никакой цели нет. Этот парадокс можно легко разрешить, если вспомнить, что в биологических системах управления функцию цели выполняет состояние другой системы, стоящей выше в биологической иерархии. Такой принцип обычно реализуется и в «рукотворных» системах иерархического управления, где цель каждого уровня «спускается» ему сверху. Этим, разумеется, не решается вопрос о том, как образована цель верхним уровнем. Поэтому проблема образования цели в иерархической постановке не решается, а переадресуется руководству. Причем у руководства возникают специфические проблемы координации целей, спускаемых подчиненным. Эти вопросы детально рассмотрены в книге [1] (пожалуй, это — первое детальное исследование иерархических систем управления), где, однако, вопрос образования цели не рассматривается.

Описанная в § 1.2 модель субъекта очень схематична и не может помочь при формализации процессов образования «хороших» целей. Она приведена лишь для того, чтобы дать представление о существовании сложных механизмов целеобразования у субъекта, о которых мы пока что мало знаем. Она впервые описана в [2].

Более подробно специфика субъекта как целеустремленной системы рассмотрена в великолепной книге [3], где без лишнего формализма доступно и в то же время глубоко проанализировано с точки зрения управления человеческое поведение как система достижения целей. Однако механизмы целеобразования и здесь не рассматриваются.

Процессам целеобразования в последнее время очень много внимания уделяют психологи, справедливо определяющие цель как первооснову поведения, которое является традиционно психологической сферой. В работе [4] механизмы целеобразования рассмотрены именно с этих позиций, но не связаны с процессами управления, рассматриваемыми в данной книге. Здесь же рассмотрен процесс формализации целей, т. е. представления их в численную форму.

1. Месарович М., Мако Д., Такахага И. Теория иерархических многоуровневых систем. — М.: Мир, 1973.
2. Граве П. С., Растринин Л. А. Об одной математической модели синтеза поведения (подсознательный уровень). — Адаптивные системы: — Рига: Зинатне, 1972, вып. 2, с. 5—15.
3. Акофф Р., Эмери Ф. О целеустремленных системах. Пер. с англ. — М.: Сов. радио, 1974.
4. Психологические механизмы целсообразования. — М.: Наука, 1977

ГЛАВА 2

ОПРЕДЕЛЕНИЕ ОБЪЕКТА УПРАВЛЕНИЯ

Задача выделения объекта управления из среды возникает после формулировки и описания множества целей управления $\{Z^*\}$. Процесс выделения объекта должен заканчиваться определением границ объекта F^0 (будем обозначать объект его оператором F^0), отделяющих его от среды, т. е. таким описанием этого объекта, которое недвусмысленно определяет его границы. В ряде случаев, когда границы объекта очевидны, такой проблемы не возникает. Это бывает, когда объект достаточно автономен (самолет, корабль, любой прибор, автомашина и т. д.). Однако в других случаях связи объекта со средой настолько сильны и разнообразны, что порой очень трудно понять, где кончается объект и начинается среда. Именно такой случай рассматривается в настоящей главе.

2.1. Анализ проблемы

Естественно возникает вопрос, а важно ли, где проводить границу раздела объекта и среды, если она нигде явно не видна? Оказывается важно и даже очень. Проиллюстрируем сказанное простым примером.

Пусть мы собираемся создать систему управления каким-то технологическим процессом. Ограничиваются ли рамки объекта только этим процессом? Ответить на этот вопрос нельзя, поскольку границы объекта управления зависят от целей, которым должен удовлетворять указанный процесс и его продукция. Если цели достаточно четкие (например, достижение определенного

качества или ритмичности работы), то неизбежно приходится управлять не только самим процессом, но и его входом, т. е. сырьем, которое перерабатывается в ходе этого процесса. Это означает, что в объект управления мы должны включить и поставщика сырья. А в некоторых случаях управлять приходится и сырьем, поступающим к поставщику. (Заметим, что необходимость в таком управлении явилась одной из причин создания комбинатов, объединений, фирм и т. д., которые имеют возможность управлять производством на всех стадиях).

Но нельзя впасть и в другую крайность, т. е. пытаться

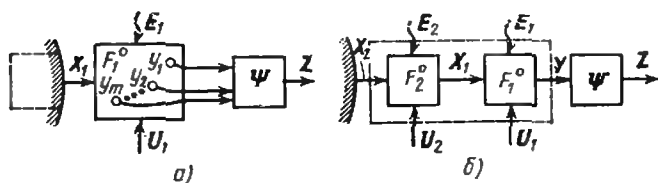


Рис. 2.1.1. К пояснению выбора границ объекта

ся управлять «всею и всею», хотя формальные основания для этого всегда найдутся, так как связи различных процессов и явлений можно проследить достаточно далеко. Существует, однако, определенный оптимум «размеров» объекта. Попробуем это показать.

На рис. 2.1.1 показаны два варианта объекта управления. Здесь $y_1, \dots, y_m$ — точки съема информации, необходимой для оценки реализуемости множества целей $\{Z^*\}$ в объекте, причем $Y = (y_1, \dots, y_m)$. Эти точки должны быть в любом варианте объекта. В первом варианте (рис. 2.1.1, а) устанавливается связь между входами и выходами в виде

$$Y = F^0_1(X_1, U_1, E_1), \quad (2.1.1)$$

где X_1 — контролируемый, U_1 — управляемый, а E_1 — неконтролируемый входы.

Этот вариант объекта можно расширить, присоединив к нему часть среды, показанную на рис. 2.1.1, б штриховой линией. Если попытаться управлять этой частью среды, т. е. считать ее частью объекта, то удастся воздействовать на ранее не управляемый вход X_1 , что, очевидно, расширяет возможности управления.

На рис. 2.1.1,б штриховой линией показан расширенный таким образом объект, для которого

$$Y = F^0_1(F^0_2(X_2, U_2, E_2), U_1, E_1). \quad (2.1.2)$$

Здесь управление U_1 не исключается, к нему добавляется еще U_2 , которое позволяет управлять дополнительно присоединенной частью.

Как видно, состояние Y у обоих вариантов объекта одинаковое, различны среды, а следовательно, и влияние среды на это состояние. Какой из вариантов лучше? Для ответа на этот вопрос нужно взвесить все «за» и «против» для обоих вариантов. Сделаем это.

Первый вариант (рис. 2.1.1,а).

«За»:

- лишь один фактор неопределенности (неконтролируемый вход) E_1 ,

- проще найти математическое описание, т. е. модель объекта F^0_1 ,

- малое число точек управления U_1 (для управления понадобится мало ресурсов R).

«Против»:

- малая управляемость, так как вход X_1 неуправляемый. Этот фактор и является решающим для введения конкурирующего второго варианта объекта.

Второй вариант (рис. 2.1.1,б).

«За»:

- возможность, хотя бы частичная, управления входом X_1 , который непосредственно воздействует на состояние Y . Это осуществляется с помощью дополнительного канала управления U_2 .

«Против»:

- расширение фактора неопределенности за счет введения дополнительной неопределенности E_2 присоединенной части,

- усложнение математического описания, так как необходимо описывать дополнительно присоединенную часть F^0_2 ,

- увеличение числа точек управления, что хотя и расширяет возможности управления, но требует и увеличения ресурсов управления R или (при его неизменности) приводит к уменьшению диапазона изменения $U = \langle U_1, U_2 \rangle$, т. е. к малым возможностям управления.

На первый взгляд может показаться, что во втором варианте «против» слишком много и нужно отдать пред-

почтение первому варианту объекта. Однако то единственное «за», которое имеет второй вариант, может оказаться решающим. Ведь нашей конечной задачей является эффективное достижение заданного множества целей $\{Z^*\}$. Второй вариант по критерию достижимости целей может оказаться лучше, чем первый.

Следовательно, задача заключается в том, чтобы суметь достаточно эффективно различать «хорошие» и «плохие» варианты выделения объектов из окружающей их среды. Как этого добиться? Если бы мы располагали формальным (математическим) описанием среды, то процесс выделения из нее объекта принципиально не представлял бы трудностей. Действительно, «высекая» различные «куски» среды и называя их различными вариантами объекта, мы всегда могли бы формально проверить, достигаются ли заданные цели управления в этом варианте объекта или нет. Повторяя эту процедуру для различных версий высекаемого, мы остановились бы на том варианте объекта, который позволяет получить максимальную (желательно 100%-ную) управляемость. Вот и все!

Но формального описания среды нет, и этот подход неприемлем. Поэтому будем искать другой подход. Направление поиска довольно очевидно. Всегда, когда нет (или пока нет) формального аппарата для решения проблемы, за решением (хотя часто и очень приближенным) обращаются к экспертам. На стадии определения объекта это единственно возможный подход. Поэтому рассмотрим процесс использования экспертов для решения поставленной задачи.

2.2. Метод экспертных оценок

Под экспертом мы будем понимать лицо, чье мнение по интересующему нас вопросу может быть более авторитетным, чем наше. В противном случае к указанному лицу не имеет смысла обращаться как к эксперту. Очевидно, что эксперт по вопросу определения объекта должен:

- хорошо знать среду, из которой выделяется объект,
- хорошо знать цели управления,
- иметь представление о возможностях и средствах управления,

— иметь представление о способах сбора и обработки информации о среде и объекте.

Эксперта не просят определять объект. Его используют как источник информации, необходимой для принятия решения о том, что же следует считать объектом управления. Это решение принимает разработчик (проектировщик) системы управления. Эксперту же задают вопросы, ответы на которые содержат требуемую информацию. Далеко не все вопросы целесообразно задавать даже эксперту (задавать-то можно, а вот его ответам верить следует с большой осторожностью).

Начнем с отбора экспертов и их квалификации. Будем квалификацию каждого i -го эксперта, т. е. его компетентность в обсуждаемом j -м вопросе, оценивать так называемым коэффициентом компетентности:

$$0 \leq k_{ji} \leq 1, \quad i=1, \dots, N_j, \quad (2.2.1)$$

где N_j — число экспертов, привлекаемых к решению j -го вопроса. При $k_{ji}=0$ i -й эксперт признается некомпетентным в j -м вопросе, а при $k_{ji}=1$ этот эксперт считается полностью компетентным. Если эти коэффициенты компетентности известны, то сразу четко определяется число экспертов. Действительно, экспертов, чья компетентность ниже пороговой, т. е.

$$k_{ji} < \delta^j, \quad (2.2.2)$$

где δ^j — заданный порог компетентности для решения j -го вопроса, можно не беспокоить решением этой проблемы. Так выясняется число N_j экспертов, привлекаемых к решению j -го вопроса (величина порога δ^j назначается проектировщиком).

Рассмотрим способ определения коэффициентов компетентности (2.2.1). Эти коэффициенты определяются также экспертно, т. е. путем взаимной оценки компетентности самих экспертов. Делается это следующим образом. Каждый i -й эксперт при решении j -го вопроса оценивает компетентность всех экспертов (в том числе и свою) с помощью коэффициентов

$$0 \leq k_{il} \leq 1, \quad l=1, \dots, N_j. \quad (2.2.3)$$

Здесь k_{il} — мнение i -го эксперта о компетентности l -го эксперта при решении j -го вопроса (k_{ii} — самооценка компетентности i -го эксперта в этом вопросе). Теперь нетрудно определить коэффициент компетентности l -го

эксперта как среднее всех оценок, включая и его собственную:

$$k^j_l = \frac{1}{N_j} \sum_{i=1}^{N_j} k^j_{li}, \quad l=1, \dots, N_j. \quad (2.2.4)$$

Заметим, что матрица взаимных оценок (2.2.3) может содержать прочерки, которые означают, что оценки нет (один эксперт не знает, а поэтому не может оценить другого). В этом случае в формуле (2.2.4) усреднение ведется по значащим оценкам, т. е. исключая прочерки.

Пример такой матрицы оценки компетентности:

l	i			
	1	2	3	4
1	1	—	0,3	0,5
2	0,7	0,8	0,4	—
3	—	0,8	0,7	0,6
4	0,7	0,7	—	0,9
k^j_l	0,8	0,77	0,47	0,67

Здесь все средние значения k^j_l оцениваются по трем значащим оценкам, т. е. $N_j=3$.

Очевидно, что этот метод самооценки экспертов не учитывает всякого рода сложные взаимоотношения, которые возможны между экспертами, как людьми. Это обстоятельство нужно иметь в виду при анализе матриц оценок компетентности. Так, если один из экспертов получил очень высокий балл у одних и очень низкий у других экспертов, то это скорее всего означает или особенности его характера, или оригинальность мышления, которые разделяются одними, но не принимаются другими экспертами. Выявление такого рода «человеческих» аномалий матрицы оценок компетентности позволит избежать ошибок, когда мнение семи компанейских «глупцов» оказывается более весомо, чем мнение одного неуживчивого «мудреца».

Располагая значениями коэффициентов компетентности (2.2.4), можно приступить к решению j -го вопроса. Всякая проблема в конечном счете сводится к определению либо некоторого числа, либо набора из q чисел, которые называют решением этой проблемы. Рассмотр-

рим первый случай, так как второй сводится обычно к q -кратному решению первого.

Пусть проблема формулируется так. «Исходя из условий (излагаются условия, где фигурирует неизвестный фактор W^j , решающий проблему), определите Ваше мнение о значении этого фактора одним числом $W^j_i = \dots$ Спасибо за экспертизу!»

Получив от всех экспертов значения W^j_i ($i=1, \dots, N_j$) (здесь могут быть и отказы от экспертизы ввиду признания собственной некомпетентности в этом вопросе; отказавшиеся эксперты автоматически выпадают из N_j), следует принять решение о значении этого фактора. Это значение с учетом компетентности экспертов естественно определять как средневзвешенное с весами, равными коэффициентам компетентности:

$$W^j = \left(\sum_{i=1}^{N_j} W^j_i k^j_i \right) \left/ \sum_{i=1}^{N_j} k^j_i \right. \quad (2.2.5)$$

Как видно, мнение каждого эксперта взвешивается значением его коэффициента компетентности. Поэтому мнение малокомпетентных экспертов автоматически учитывается в меньшей степени. Следовательно, экспертов можно не отбирать по критерию (2.2.2). Этот критерий позволяет лишь не беспокоить экспертов, чье мнение несущественно для решения j -го вопроса.

Формула (2.2.5) и решает поставленную задачу. Однако здесь следует учесть и принять во внимание следующее важное обстоятельство. Мнения экспертов, т. е. числа W^j_i могут сильно отличаться друг от друга по двум причинам. С одной стороны, разноречивой может быть вызван тем, что экспертам трудно отвечать на поставленный вопрос. В этом случае следует изменить его постановку, чтобы облегчить экспертам их задачу. С другой стороны это может произойти и потому, что по поводу фактора W^j существуют две противоречащие друг другу точки зрения, которые следует учитывать при решении вопроса. В этом случае проектировщику следует выявить их и присоединиться к одной из них.

Разброс мнений проще всего определять с помощью оценки дисперсии полученных экспертных оценок:

$$\hat{D}(W^j) = \frac{1}{N_j - 1} \sum_{i=1}^{N_j} \left(W^j_i - \frac{1}{N_j} \sum_{i=1}^{N_j} W^j_i \right)^2 \quad (2.2.6)$$

Малая величина этой дисперсии свидетельствует о достаточном единодушии экспертов.

Как видно, метод экспертных оценок позволяет решать формально-неразрешимые проблемы. Правда, полученное решение всегда приближенно, но его можно уточнять, увеличивая число экспертов и учитывая их компетентность в решении этой проблемы.

Теперь применим изложенный метод к решению поставленной задачи выделения объекта из среды.

2.3. Управляемость объекта

Процедуру выделения объекта из среды естественно строить так, чтобы в получившемся объекте реализовались цели управления, т. е. чтобы им можно было бы эффективно управлять в ситуациях, которые складываются в процессе управления. Для этого введем понятие *управляемости* объекта, под которым будем подразумевать вероятность достижения задаваемых целей в различных ситуациях.

Прежде всего рассмотрим понятие ситуации. Под ситуацией, сложившейся в процессе управления, в самом общем случае следует понимать тройку

$$S = \langle X, E, Z^* \rangle, \quad (2.3.1)$$

определяющую состояние неуправляемых входов объекта и цель, которую следует достигнуть (легко заметить, что это определение ситуации обобщает предложенное ранее (1.3.12), где управление U зависит в конечном счете лишь от X и Z^*). Все ситуации, которые встречаются в процессе управления данным объектом, можно подразделить на два подмножества ситуаций — управляемых, при которых заданная цель Z^* всегда достигается, и неуправляемых, когда эта цель Z^* не достигается.

Обозначим $\{S\}$ множество всех возможных ситуаций S , встречающихся в процессе управления. Пусть $\{J_S\}$ — подмножество ситуаций $\{S\}$, где объект неуправляем, т. е. не все цели из $\{Z^*\}$ достигаются, а $\{J_S\}$ — подмножество ситуаций, где объект управляем, т. е. все цели из $\{Z^*\}$ достигаются. Очевидно, что

$$\{J_S\} \cup \{J_{\bar{S}}\} = \{S\}. \quad (2.3.2)$$

Это выражение означает очевидное: каждая ситуация S может быть или управляемой, или неуправляемой.

Каждому элементу множества $\{S\}$, т. е. каждой ситуации S , поставим в соответствие число $\rho = \rho(S)$, которое определяло бы вероятность появления этой ситуации S . Если число элементов множества $\{S\}$ бесконечно, то под $\rho(S)$ следует, как обычно, понимать плотность этой вероятности. В первом (дискретном) случае

$$\sum_{i=1}^N \rho(S_i) = 1,$$

где N — общее число встречающихся ситуаций. Во втором (непрерывном) случае

$$\int_{\{S\}} \rho(S) dS = 1, \quad (2.3.3)$$

где интеграл берется по всему множеству $\{S\}$. Соотношение (2.3.3) выражает то обстоятельство, что вне области $\{S\}$ не могут встретиться реальные ситуации S .

Непрерывный случай более удобен и здесь обобщает дискретный, поэтому будем пользоваться только им, хотя для практических целей и придется вернуться к дискретному.

Теперь уточним понятие управляемости. *Управляемостью объекта* будем называть вероятность того, что случайно выбранная ситуация S управляема. Эта вероятность равна

$$P = \int_{\{J_S\}} \rho(S) dS, \quad (2.3.4)$$

где интеграл берется по подмножеству управляемых ситуаций $\{J_S\}$.

Неуправляемость определяется аналогично как интеграл по множеству $\{J_{\bar{S}}\}$ неуправляемых ситуаций:

$$\bar{P} = \int_{\{J_{\bar{S}}\}} \rho(S) dS. \quad (2.3.5)$$

Из (2.3.2) и (2.3.3) следует, что

$$P + \bar{P} = 1. \quad (2.3.6)$$

Если из двух рассматриваемых вариантов объекта один обладает большей управляемостью, чем другой, то

именно этот объект при прочих равных условиях следует предпочитать другому. Следовательно, задача выбора объекта сводится к задаче определения его управляемости P .

Однако строгое определение множеств $\{J_S\}$ и $\{J_{\bar{S}}\}$, а также функции $\rho(S)$ невозможно из-за отсутствия формального описания среды. Именно поэтому для решения этой задачи привлекаются эксперты.

Для определения (точнее, оценки) управляемости P объекта необходимо декомпозировать проблему, т. е. представить ее в виде ряда более простых задач, которые могут быть решены экспертами.

Для этого экспертам необходимо предоставить следующие исходные данные:

1) цели управления, т. е. достаточно полное описание множества целей управления $\{Z^*\}$, которые должны быть реализованы в объекте,

2) сведения о располагаемых ресурсах R , которые выделены на создание системы управления и ее эксплуатацию.

Обозначим эти исходные данные двойкой:

$$A = \langle \{Z^*\}, R \rangle. \quad (2.3.7)$$

Экспертам должна быть предоставлена также полная информация о предполагаемом объекте и его связях со средой, т. е.:

3) описание объекта F^o ,

4) описание множества $\{X\}$ контролируемых состояний среды, которые могут быть учтены в процессе синтеза управления («объем» этого множества, очевидно, зависит от ресурсов R , выделенных для организации системы контроля и измерения X),

5) описание $\{U\}_R$ — множества управлений, допустимых ресурсами R (имеющейся энергией, временем, объемом памяти управляющего устройства, скоростью и спецификой переработки информации и т. д.),

6) описание $\{E\}$ — множества неконтролируемых факторов среды и объекта.

Эти данные об объекте обозначим четверкой:

$$B = \langle F^o, \{X\}, \{U\}_R, \{E\} \rangle. \quad (2.3.8)$$

Эксперт, располагая указанными исходными данными A и B , должен оценить управляемость P объекта. Обо-

значив через φ алгоритм работы эксперта, в результате можно записать

$$P = \varphi(A, B). \quad (2.3.9)$$

Однако, как легко представить, в этой ситуации эксперту работать очень трудно. Действительно, информация A и B хотя и содержит необходимые сведения об управляемости P объекта, но оценить управляемость P , т. е. реализовать алгоритм φ , экспертам не удастся. Эта задача слишком для них сложна. Поэтому еще раз декомпозируем задачу. Для этого введем различные формы управляемости и неуправляемости объекта.

Будем называть объект *абсолютно управляемым* ($P=1$), если каждая ситуация из $\{S\}$ управляема, т. е. цель всегда достигается. Это означает, что для любого контролируемого состояния среды $X \in \{X\}$, любого неконтролируемого входа $E \in \{E\}$ и для любой цели $Z^* \in \{Z^*\}$ всегда найдется такое управление $U^* \in \{U\}_R$, которое переведет объект в требуемое состояние, т. е.

$$Z = Z(X, U^*, E) = \Psi(F^0(X, U^*, E)) = Z^*, \quad (2.3.10)$$

где $\Psi(\cdot)$ — преобразование пространства состояний объекта $\{Y\}$ в пространство целей $\{Z\}$, рассмотренное в (1.3.6); Z — состояние объекта в пространстве целей; $Z(\cdot, \cdot, \cdot)$ — функция, характеризующая зависимость Z от X , U и E .

Для экономной записи этого определения удобно воспользоваться так называемыми кванторами существования ($\exists$) и всеобщности ($\forall$). Тогда

$$\forall X \in \{X\}, \forall E \in \{E\}, \forall Z^* \in \{Z^*\}, \\ \exists U^* \in \{U\}_R : Z(X, U^*, E) = Z^*. \quad (2.3.11)$$

Это на первый взгляд сложное выражение читается очень просто: для любого X , принадлежащего множеству $\{X\}$, что записывается в виде $\forall X \in \{X\}$, любого E , принадлежащего множеству $\{E\}$ (т. е. $\forall E \in \{E\}$), и любого Z^* , принадлежащего $\{Z^*\}$ (т. е. $\forall Z^* \in \{Z^*\}$), существует такое U^* , принадлежащее $\{U\}_R$ (т. е. $\exists U^* \in \{U\}_R$), при котором $Z = Z^*$, т. е. цель достигается.

Абсолютная управляемость ($P=1$) редко встречается на практике при управлении сложными объектами. Действительно, для ее обеспечения следует или значительно сужать множество целей, или иметь постоянную среду ($X = \text{const}$), или выделять очень большие ресур-

сы R на управление. Как правило, абсолютно управляемы либо простые объекты (например, системы автоматического регулирования), либо очень ответственные (например, системы обеспечения жизнеспособности операторов в экстремальных условиях). Подавляющее большинство систем управления сложными объектами нельзя отнести к этому типу.

Поэтому следует ввести понятие *частичной* или *относительной управляемости*. Будем различать причины неуправляемости. Для этого введем типы неуправляемости следующим образом. X — неуправляемость связана со спецификой множества $\{X\}$, в котором нашлось такое X , что при любом $E \in \{E\}$ не нашлось такого $U \in \{U\}_R$, при котором $Z = Z^*$, т. е. любая цель не достигается. Определим X -неуправляемость формально:

$$J_{\bar{X}}: (\exists X \in \{X\}: \forall E \in \{E\}, \forall Z^* \in \{Z^*\}, \\ \bar{\exists} U \in \{U\}_R: Z(X, U, E) = Z^*). \quad (2.3.12)$$

Здесь $J_{\bar{X}}$ — X -неуправляемость и использовано отрицание квантора существования $\bar{\exists}$, что означает «не существует» (не найдется). Выражение (2.3.12) читается следующим образом: X -неуправляемость определяется как существование такого допустимого X , при котором для любых допустимых E и Z^* не найдется такого допустимого U , при котором цель Z^* будет достигнута.

Приведем пример такой ситуации. Пусть X^* — энергия, требуемая для нормального функционирования объекта. Тогда при $X < X^*$, т. е. при недостаточной поставке энергии, объект не будет работать и не выполнит возлагаемые на него цели Z^* несмотря на все усилия U , предпринятые в рамках выделенных ресурсов R .

E -неуправляемость вводится аналогично, как существование таких неконтролируемых состояний среды, при которых при любых допустимых X не удастся достигнуть любой допустимой цели:

$$J_{\bar{E}}: (\exists E \in \{E\}: \forall X \in \{X\}, \\ \forall Z^* \in \{Z^*\}, \bar{\exists} U \in \{U\}_R: Z(X, U, E) = Z^*). \quad (2.3.13)$$

Пример. Пусть E — амортизация оборудования, которая приводит к снижению точности приборов. Очевидно, что начиная с некоторого уровня амортизации потеря точности приборов будет столь

велика, что не позволит с их помощью реализовать заданные цели, например выпускать качественную продукцию даже при специальных мероприятиях U , проводимых в рамках ресурсов R .

Z^* -неуправляемость характеризуется наличием таких допустимых целей, которые при прочих допустимых условиях никогда не достигаются:

$$J_{Z^*} : (\exists Z^* \in \{Z^*\} : \forall X \in \{X\},$$

$$\forall E \in \{E\}, \exists U \in \{U\}_R : Z(X, U, E) = Z^*). \quad (2.3.14)$$

Это означает, что существуют допустимые цели, которые достижимы не при всех состояниях среды. Напри-

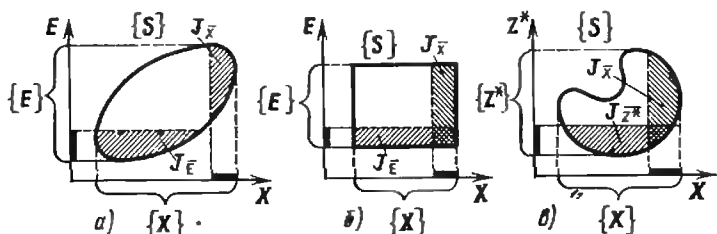


Рис. 2.3.1. Зоны неуправляемости объекта, вызванной факторами X и E (а и б) и факторами Z^* и X (в).

мер, цель перевыполнения плана не реализуется при нерегулярных поставках (X) или аварийных ситуациях (E), которые допускают выполнение плана.

На рис. 2.3.1 для иллюстрации показано несколько случаев неуправляемости в пространстве состояний $\{S\} = \{X, E, Z^*\}$. Жирными отрезками обозначены зоны неуправляемости, которые порождают области неуправляемости J_X , J_E и J_{Z^*} . Как видно, эти зоны могут

пересекаться (см. рис. 2.3.1, б, в). Отметим это обстоятельство — мы будем использовать его в дальнейшем.

Теперь введем понятия перекрестных неуправляемостей, т. е. таких, которые зависят сразу от двух или трех факторов X, E, Z^* , определяющих успех управления:

XE -неуправляемость определяется как одновременное существование таких X и E , при которых не всякая цель достижима:

$$J_{XE} : (\exists (X, E) : \forall Z^*, \exists U \in \{U\}_R : Z(X, U, E) = Z^*). \quad (2.3.15)$$

Например, одновременное снижение качества сырья (X) и амортизация оборудования (E) ставят обычно под угрозу выполнение плана, хотя каждый из этих факторов в отдельности может быть преодолен соответствующим управлением U .

Здесь и далее для сокращения записи предполагается, что $X \in \{X\}$, $E \in \{E\}$, $Z^* \in \{Z^*\}$, $U \in \{U\}_R$.

XZ^* -неуправляемость определяется аналогично:

$$J_{XZ^*}^- : (\exists (X, Z^*) : \forall E, \exists U : Z(X, U, E) = Z^*); \quad (2.3.16)$$

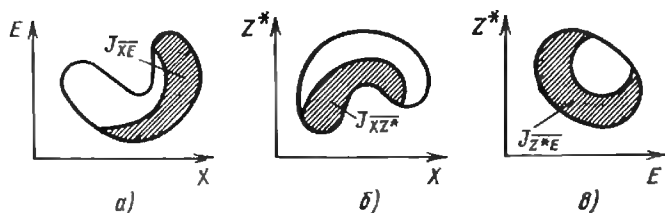


Рис. 2.3.2. Зоны неуправляемости объекта, вызванной парными взаимодействиями факторов X , E и Z^*

Z^*E -неуправляемость:

$$J_{Z^*E}^- : (\exists (Z^*, E) : \forall X \exists U : Z(X, U, E) = Z^*). \quad (2.3.17)$$

И, наконец, XEZ^* -неуправляемость:

$$J_{XEZ^*}^- : (\exists (X, E, Z^*) : \exists U : Z(X, U, E) = Z^*). \quad (2.3.18)$$

На рис. 2.3.2 зоны неуправляемости такого вида заштрихованы. Как видно, они включают и зоны, рассмотренные на рис. 2.3.1. Однако не следует путать области неуправляемости такого вида с пересечением областей X -, E - и Z^* -неуправляемости, показанных на рис. 2.3.1, б и в.

Если бы с помощью экспертов удалось оценить встречаемость (вероятность появления) всех указанных видов неуправляемости, то процесс определения управляемости объекта свелся бы к вычислению суммы

$$P = 1 - \sum_{i=1}^7 p_i, \quad (2.3.19)$$

где p_i — вероятность появления i -го вида неуправляемости (всего их 7). Это и был бы алгоритм Φ в (2.3.9). Располагая таким алгоритмом, можно из определенного конечного множества объектов-претендентов выбрать один, наиболее управляемый.

Теперь снова обратимся к экспертам, они нам нужны для оценок величин p_i ($i=1, \dots, 7$), входящих в оценку управляемости (2.3.19).

2.4. Экспертный метод оценки управляемости объекта

Итак, следует сформулировать вопросы отобранным N экспертам таким образом, чтобы их ответы позволили принять наиболее достоверное решение об управляемости объекта в предложенных границах. Для этого, прежде всего, необходимо образовать конечное число альтернатив — вариантов объекта $F^0_1, \dots, F^0_q$, т. е. описаний:

$$Y = F^0_v(X_v, U_v, E_v), \quad v = 1, \dots, q, \quad (2.4.1)$$

где X_v , U_v и E_v — соответственно наблюдаемые, управляемые и ненаблюдаемые входы v -го варианта объекта. Образование вариантов (2.4.1) производится исходя из предварительного анализа среды и целей.

Таким образом, задача определения объекта сводится к выбору одной из альтернатив (2.4.1). Исходная информация для этого выбора содержится в ответах экспертов на вопросы, с помощью которых определяется управляемость каждого варианта объекта (2.4.1). Будем задавать экспертам такие вопросы, чтобы по ответам определить вероятность каждого вида неуправляемости. Для этого выявим ситуации S , при которых цели не достигаются, т. е. объект неуправляем. Эти ситуации естественно классифицировать по видам неуправляемости.

Экспертиза проходит в несколько этапов. Сначала эксперты выявляют и описывают неуправляемые ситуации. Затем составляется полный перечень таких ситуаций, объединенных по виду неуправляемости. Этот перечень предлагают экспертам для оценки вероятности каждой ситуации. Если эти оценки не слишком различны, то они принимаются для вычисления управляемости объекта. Однако при большом разном экспертам

предоставляется возможность изменить свою оценку, учитывая оценки других экспертов по спорным ситуациям. Это делается до тех пор, пока оценки перестанут сильно отличаться. По результатам этих оценок вычисляется управляемость. Рассмотрим каждый этап в отдельности.

Итак, на первом этапе каждому эксперту предлагается описать все неуправляемые ситуации по каждому виду неуправляемости:

$$J_{\bar{X}}:(S^1_1, \dots, S^1_{l_1}), \dots, J_{\overline{XEZ}^*}:(S^1_7, \dots, S^1_{l_7}), \quad (2.4.2)$$

где S^j_i — j -я неуправляемая ситуация i -го вида неуправляемости.

Очевидно, у каждого эксперта будет свой набор таких ситуаций. Эти ситуации объединяются в общий список неуправляемых ситуаций. Он имеет тот же вид, что и (2.4.2), но с другими числами $l_1, \dots, l_7$.

При составлении этого списка следует следить, чтобы ситуации в списке не повторялись. Это необходимо, так как, например, ситуации из $J_{\overline{XE}}$ можно спутать с ситуациями, одновременно принадлежащими $J_{\bar{X}}$ и $J_{\bar{E}}$ (ср. пересечение областей на рис. 2.3.1, б и в и 2.3.2, а и б).

Далее каждому эксперту предлагается полный список всех неуправляемых ситуаций

$$S_1, \dots, S_n \quad (2.4.3)$$

без учета принадлежности ситуации виду неуправляемости (это нужно было лишь для выявления ситуаций (2.4.3) экспертами). Каждый j -й эксперт каждой ситуации S_i приписывает число

$$0 \leq g^j_i, \quad i=1, \dots, n; \quad j=1, \dots, N, \quad (2.4.4)$$

характеризующее, по его мнению, оценку вероятности появления такой ситуации в процессе управления этим объектом. При этом, очевидно, должно выполняться ограничение

$$\sum_{i=1}^n g^j_i \leq 1, \quad j=1, \dots, N.$$

Заметим, что сумма

$$\bar{P}_j = \sum_{i=1}^n g^j_i, \quad j=1, \dots, N \quad (2.4.5)$$

является мнением j -го эксперта о неуправляемости объекта.

Однако, прежде чем определять среднюю неуправляемость объекта, следует проверить согласованность экспертов. Оценить эту согласованность можно по дисперсиям индивидуальных оценок различных ситуаций

$$\hat{D}_i = \frac{1}{N-1} \sum_{j=1}^N \left(g^j_i - \frac{1}{N} \sum_{j=1}^N g^j_i \right)^2, \quad i=1, \dots, n. \quad (2.4.6)$$

Если найдутся такие ситуации S_i , о вероятности появления которых мнения экспертов сильно расходятся, т. е.

$$D_i > \delta, \quad i=1, \dots, n, \quad (2.4.7)$$

где $\delta > 0$ — некоторый заданный порог, то по этим ситуациям экспертам предлагается пересмотреть или подтвердить свою оценку с учетом мнений других экспертов (эти ситуации и мнения всех экспертов по этим ситуациям сообщаются). Процедура повторяется до тех пор, пока не выполнится условие

$$D_i \leq \delta, \quad i=1, \dots, n.$$

Теперь экспертная оценка неуправляемости объекта принимает вид

$$\bar{P} = \left(\sum_{j=1}^N k_j \bar{P}_j \right) / \sum_{j=1}^N k_j, \quad (2.4.8)$$

где k_j — компетентность j -го эксперта (2.2.5), а $\bar{P}_j$ — оценка неуправляемости объекта j -м экспертом (2.4.5).

На этом заканчивается экспертная оценка неуправляемости данного варианта объекта.

Получив подобные экспертные оценки для всех q вариантов (2.4.1) объекта, найдем оценки их неуправляемости $\bar{P}_1, \dots, \bar{P}_q$. Очевидно, что наилучший вариант объекта тот, который имеет минимальную управляемость. Если

$$\bar{P}_l = \min_{v=1, \dots, q} \bar{P}_v,$$

то l -й вариант объекта и следует считать оптимальным, т. е. $F^0 = F^0_l$.

Естественно, что при этом следует учитывать дисперсию оценок вероятностей $\bar{P}_v (v=1, \dots, q)$. Если несколько вариантов объекта окажутся статистически неразличимы, то либо следует увеличить число экспертов, либо сообщить им дополнительную информацию о среде, целях и ресурсах, либо ввести дополнительные критерии оценки вариантов объекта, по которым можно осуществить однозначный выбор.

ЛИТЕРАТУРНЫЙ КОММЕНТАРИЙ К ГЛАВЕ 2

Вторичность объекта по отношению к цели связана прежде всего с тем, что объект является системой, которая в соответствии с [1, 2] представляет собой множество элементов, связанных отношением, обладающим заданным свойством. Это свойство и определяется множеством целей, которые должны быть реализованы в данном объекте. Таким образом, объект управления отличается от системы вообще тем, что он образован элементами, связанными отношением, обладающим свойством иметь возможность объекту находиться в состояниях из заданного целевого множества. Указанная необходимость состояниям объекта соответствовать целевому множеству и образует зависимость объекта от целей управления.

Метод экспертных оценок, рассмотренный в § 2.2, как видно, позволяет получать ответы на вопросы, которые не могут быть получены формально хотя бы потому, что вопросы не поставлены формально. Таких вопросов в наше время задается значительно больше, чем хочется сторонникам формализма. Дело в том, что сейчас все чаще и чаще приходится управлять объектами задолго до их исчерпывающего изучения и зачастую переходить к управлению другими, так и не узнав достаточно предыдущие. В такой обстановке естественно, что только экспертный метод позволяет получить ответы на поставленные вопросы. При этом приходится часто «расплачиваться» высокой дисперсией ответов, что, однако, не должно смущать, так как в настоящее время разработаны эффективные методы принятия решения в стохастической обстановке (см., например, вполне доступные книги по теории стохастических решений [3 или 4]).

Таким образом, использование экспертных решений вовсе не является «заматанием мусора под коврик», как это может показаться. В сочетании с методами теорий статистических решений экспертный метод дает вполне приемлемые результаты.

По методу экспертных оценок имеется ряд полезных руководств. Отметим среди них [5, 6]. Для начального ознакомления можно рекомендовать две главы из доступной и обстоятельной книги [7, с. 143—260].

Введенное в § 2.3 понятие управляемости объекта отличается от известного ранее понятия управляемости динамической системы (см., например, [8]), которое подразумевает существование управления, переводящего объект в требуемое состояние. Предложенное понятие, как видно, обобщает известное на стохастический случай. Поэтому, по-видимому, правильнее было бы назвать введенную характеристику объекта статистической (или вероятностной) управляемостью, что, однако, привело бы к слишком громоздкой терминологии.

- 1 Уемов А. И. Вещи, свойства и отношения. — М.: Изд-во АН СССР, 1963.
- 2 Уемов А. И. Системы и системные параметры. — В кн.: Проблемы формального анализа систем. — М.: Высшая школа, 1968, с 15—35.
- 3 Чернов Г., Мозес Л. Элементарная теория статистических решений. — М.: Сов. радио, 1962.
- 4 Де Гроот М. Оптимальные статистические решения: Пер. с англ. — М.: Мир, 1974.
- 5 Бешелев С. Д., Гурович Ф. Г. Математико-статистические методы экспертных оценок. — М.: Статистика, 1974.
- 6 Экспертные оценки в научно-техническом прогнозировании. — Киев: Наукова думка, 1974.
- 7 Шляпентох В. Как сегодня изучают завтра. — М.: Сов. Россия, 1975.
- 8 Медич Д. Статистические оптимальные линейные оценки и управление. — М.: Энергия, 1973.

ГЛАВА 3

СТРУКТУРНЫЙ СИНТЕЗ МОДЕЛИ
ОБЪЕКТА

Структура модели объекта управления определяет вид и характер связи между входами (X и U) и выходами (Y) модели объекта независимо от конкретных значений параметров, которые определяются на последующих этапах управления — этапах идентификации (см. гл. 4) и планирования экспериментов (см. гл. 5). Процесс структурного синтеза модели объекта разобьем на следующие стадии:

- определение входов и выходов;
- экспертное ранжирование входов и выходов;
- декомпозиция модели;
- выбор структурных элементов модели.

Рассмотрим каждую стадию в отдельности.

3.1. Определение входов и выходов объекта

Для определения структуры модели как многополюсника (т. е. преобразователя типа «черного ящика» со многими входами и выходами) необходимо выяснить, какие именно входы и выходы объекта будут включены в его модель. Для этого прежде всего выявляют всех воз-

можных «претендентов» на роль входов и выходов и из них выделяют наиболее существенные, которые и образуют многополюсник модели.

Но здесь сразу возникает вопрос о том, какие факторы в этом случае называть существенными. Ответ однозначный: так как модель объекта создается для целей управления, то существенными являются те факторы, которые наиболее сильно влияют на осуществление целей управления в объекте. Это связано с тем, что при формировании структуры системы управления прежде

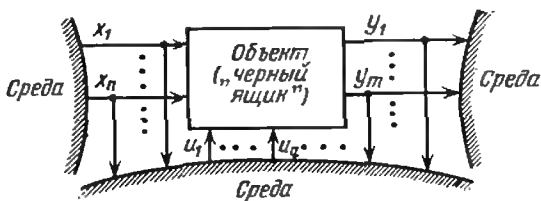


Рис. 3.1.1. Схема взаимодействия объекта («черного ящика») со средой

всего необходимо знать, какие воздействия будет (и может) испытывать объект управления и как результаты этих воздействий связаны с достижением целей управления объектом.

Объект связан со средой, вообще говоря, бесконечным числом связей (рис. 3.1.1), определяющих его состояние. При синтезе модели должны быть выявлены наиболее сильные, существенные связи, чтобы, отбросив слабые связи, получить минимальное число взаимодействий объекта со средой. Однако прежде чем осуществить такое подразделение, необходимо перечислить все связи. Очевидно, что это только наблюдаемые связи, так как ненаблюдаемые не включаются в модель. Будем подразделять их на три типа:

— неуправляемые, но контролируемые связи (X), характеризующие воздействие среды на объект (из них образуются неуправляемые входы);

— управляемые связи (U), с помощью которых можно целенаправленно изменять состояние объекта (из них образуются управляемые входы);

— информирующие связи (Y), позволяющие определить состояние объекта (из них образуются выходы объекта).

Рассмотрим каждый тип связи отдельно.

Неуправляемые связи X должны удовлетворять следующим требованиям.

Во-первых, они должны влиять на реализацию целей $\{Z^*\}$ в объекте управления. Для оценки этого влияния можно произвести следующий мысленный эксперимент. Пусть объект находится в состоянии Y^* , которое соответствует заданной цели Z^* , т. е. цель достигнута (неважно каким образом, например с помощью какого-то необходимого для этого случая управления U^*). Пусть далее параметр x_i , характеризующий состояние i -й связи объекта со средой, изменяется, т. е. изменяется среда (но только ее i -я связь с объектом). Если это изменение среды может изменить состояние объекта настолько, что цель Z^* не достигается, то связь x_i существенна для управления объектом, так как она влияет на процесс реализации целей в объекте. В противном случае этой связью можно пренебречь.

Во-вторых, эти связи должны эффективно и просто измеряться. Если параметр x_i не измеряется, т. е. не существует методов ее измерения или они слишком сложны или дороги, то эту связь нельзя учесть при синтезе управления и она должна рассматриваться как случайная помеха ϵ .

Таким образом, для отбора факторов среды, которые должны учитываться при управлении, необходима их существенность (важность) для реализации целей управления в объекте и эффективная измеримость (контролируемость).

Ввиду отсутствия моделей среды и объекта процесс отбора учитываемых факторов среды производится экспертным методом, т. е. с привлечением специалистов (экспертов), хорошо знающих объект и среду. При этом уровень осведомленности экспертов определяет эффективность процесса выяснения существенных связей объекта со средой. Пусть по этому принципу отобраны n связей:

$$x_1, \dots, x_n. \quad (3.1.1)$$

Это значит, что содержательно определена каждая связь (например, x_1 — температура среды, x_2 — атмосферное давление, x_3 — напряжение и т. д.) и установлено, что эти связи как-то влияют на реализацию целей в объекте.

Теперь рассмотрим процесс определения управляе-

мых связей U — каналов будущего управления. Здесь важнейшими критериями отбора являются:

— воздействие связи на состояние объекта, т. е. возможность компенсировать негативные изменения этого состояния, вызванные различными факторами как среды X , так и самого объекта F^0 ,

— управляемость, т. е. возможность оперативно без существенных затрат изменять состояние U каналов воздействия на объект. По этим критериям также экспертным путем содержательно определяются связи

$$u_1, \dots, u_q, \quad (3.1.2)$$

(например, u_1 — расход подаваемого топлива, u_2 — команда на включение какого-то агрегата и т. д.).

И, наконец, информирующие связи Y . Они должны удовлетворять двум естественным требованиям:

— нести информацию о выполнении (или невыполнении) целей в объекте управления,

— надежно и оперативно контролироваться (измеряться).

Выделенные таким образом информирующие связи

$$y_1, \dots, y_m \quad (3.1.3)$$

заканчивают описание объекта как «черного ящика», т. е. как многополюсника с неизвестной (пока) внутренней структурой (например, y_1 — температура объекта, y_2 — скорость его движения и т. д.).

Таким образом, этап определения входов и выходов заключается в содержательном описании всех тех контролируемых входов и выходов объекта, независимо от его внутренней структуры, которые связаны с реализацией целей в объекте. Очевидно, что их число должно быть больше, чем будет реализовано в модели объекта. Такая избыточность необходима для осуществления эффективного отбора наиболее существенных факторов (связей) на следующем этапе.

3.2. Экспертное ранжирование входов и выходов

Выделенные в предыдущем параграфе связи еще не являются входами и выходами модели. Это лишь «претенденты». Входами и выходами они станут после того, как будут выделены наиболее существенные из них и

отброшены несущественные. Действительно, модель во-
все не должна отражать абсолютно все стороны объекта,
т. е. не должна (да и не может) быть во всем адекватной
объекту. Для целей управления вполне достаточно вы-
явить ее наиболее существенные связи со средой, необ-
ходимые для управления, т. е. она может быть прибли-
женной. Поэтому процессу выбора наиболее существен-
ных факторов предшествует процедура ранжирования
всех их по признаку существенности для реализации це-
лей управления.

Рассмотрим процесс ранжирования факторов x_i ($i=$
 $=1, \dots, n$). Ранжирование факторов U и Y происходит
аналогично.

Прежде всего, определим понятие ранжирования. Под
ранжированием будем понимать процедуру расположе-
ния факторов x_i в порядке их существенности: на пер-
вом месте стоит самый существенный, следом за ним
менее существенный, но самый важный из оставшихся
и т. д. Полученный таким образом ранжированный ряд
имеет вид

$$x_{i_1}, x_{i_2}, \dots, x_{i_n}, \quad (3.2.1)$$

где i_1 — номер самого существенного фактора, i_2 — номер
менее существенного и т. д. до i_n — номера самого несущ-
ественного фактора в этом ряду.

Однако можно, а иногда и удобнее поступать иначе.
Каждому фактору x_i поставить в соответствие некоторое
целое число — его ранг k_i , т. е. номер фактора в ранжи-
рованном ряду (3.2.1):

$$\begin{aligned} x_1, x_2, \dots, x_n, \\ k_1, k_2, \dots, k_n. \end{aligned} \quad (3.2.2)$$

Очевидно, что первый ранг ($k_i=1$) имеет вход x_{i_1} ,
наиболее влияющий на реализацию цели в объекте. Вто-
рой и следующие ранги ($k_i=2$ и т. д.) в порядке убыва-
ния их важности имеют входы, влияние которых не столь
существенно. Таким образом, связь ранжированного
ряда (3.2.1) и рангов (3.2.2) определяется простой и оче-
видной формулой

$$k_{i_j} = j, \quad i, j = 1, \dots, n, \quad (3.2.3)$$

т. е. ранг фактора с номером i_j равен j .

Например, если ранги (3.2.2) оказались равными

$$\begin{aligned}x_i &= x_1, & x_2, & x_3, & x_4, & x_5, \\k_i &= 3, & 1, & 5, & 4, & 2,\end{aligned}\tag{3.2.2a}$$

то ранжированный ряд имеет вид x_2, x_5, x_1, x_4, x_3 . Действительно, из (3.2.2a) видно, что первый ранг ($k_i=1$) имеет второй фактор, второй — пятый и т. д. Теперь, если придется создавать систему управления с ограниченной информацией о среде ($n=3$), выбор существенных факторов из (3.2.2a) очевиден. Это x_2, x_5 и x_1 . Четвертым и третьим факторами при этом пренебрегаем, причем очевидно, что ущерб от этого решения будет минимальным, так как отброшены самые несущественные факторы.

Задача построения ранжированного ряда (3.2.1) или эквивалентная ей задача определения рангов (3.2.2) решается экспертами и сводится к организации экспертного опроса и обработки результатов этого опроса, с тем чтобы получить искомые ранги и оценить их достоверность, т. е. согласованность экспертов.

Рассмотрим два метода экспертного ранжирования: непосредственного ранжирования и парных сравнений. В первом методе эксперты сразу присваивают ранги факторам, которые им представлены для ранжирования, во втором — используется парное сравнение факторов, что упрощает задачу эксперта, но требует дальнейшей обработки результатов для получения ранжированного ряда.

Метод непосредственного ранжирования. Пусть N экспертов ранжируют n факторов $x_1, \dots, x_n$. Каждому фактору каждый эксперт присваивает ранг — целое число от 1 до n . Так, i -му фактору (x_i) j -й эксперт ($\mathcal{E}_j$) присваивает ранг k_{ij} . В результате получается матрица мнений экспертов размерностью $N \times n$

$$\begin{array}{c} \mathcal{E}_1 \\ \vdots \\ \mathcal{E}_N \end{array} \left\| \begin{array}{cccc} x_1 & x_2 & \dots & x_n \\ k_{11} & k_{21} & \dots & k_{n1} \\ \dots & \dots & \dots & \dots \\ k_{1N} & k_{2N} & \dots & k_{nN} \end{array} \right\|, \tag{3.2.4}$$

где номера строк соответствуют номерам экспертов, а номера столбцов — номерам ранжируемых факторов, т. е. j -я строка представляет собой мнение j -го эксперта $\mathcal{E}_j$, а i -й столбец — мнение всех экспертов по поводу i -го фактора (x_i).

При назначении рангов экспертами нужно соблюдать следующие условия:

— во-первых, сумма рангов, назначенных всем факторам каждым экспертом, должна быть одинакова. Это

означает, что сумма элементов любой строки матрицы равна

$$\sum_{i=1}^n k_{ij} = \sum_{i=1}^n i = \frac{n(n+1)}{2}, \quad j=1, \dots, N;$$

— во-вторых, если эксперт какие-то q факторов считает эквивалентными или одинаковыми по важности, то он присваивает им один ранг. Этот ранг равен среднему из q целых рангов, таких, которые получились бы при условии, что эксперту удалось их проранжировать. Например, эквивалентность четырех факторов ($q=4$): x_1 , x_2 , x_3 , x_6 , стоящих на пятом месте в ранжированном ряду, приводит к равенству их рангов:

$$k_1=k_2=k_3=k_6=(5+6+7+8)/4=6,5.$$

Как видно, в этом случае ранги могут быть дробными.

Теперь, для окончательного определения искомых рангов следует вычислить средние ранги каждого фактора:

$$\bar{k}_i = \frac{1}{N} \sum_{j=1}^N k_{ij}.$$

Эти ранги и позволяют проранжировать факторы. На первом месте ставится фактор, имеющий минимальный средний ранг,

$$\bar{k}_l = \min_{i=1, \dots, n} \{\bar{k}_i\},$$

т. е. фактор x_l , на втором — фактор, имеющий минимальный из остальных средний ранг, и т. д. Полученные ранги позволяют построить ранжированный ряд факторов, который и будет соответствовать усредненному мнению коллектива N экспертов.

Очевидно, что далеко не всякий результат экспертного опроса следует считать удовлетворительным. Действительно, если мнения экспертов сильно расходятся (например, одна половина экспертов фактору x_i присвоила первый ранг, а другая — последний), то такое ранжирование не может быть положено в основу решения о выборе существенных факторов. Поэтому для оценки результатов всякого экспертного опроса вводится критерий, характеризующий согласованность экспертов. Чем

выше эта согласованность, тем в большей степени можно «верить» результатам экспертного опроса, и наоборот.

Согласованность экспертов удобно определять как степень рассеяния средних рангов $\bar{k}_i$ ($i=1, \dots, n$). Действительно, если мнения экспертов полностью совпадают, что средние ранги представляют собой целые, не равные друг другу числа (случай одинаковых рангов здесь пока не рассматривается). Графически это изображено на рис. 3.2.1, а, где точками показано расположение средних рангов на числовой оси. Если же мнение экспертов пол-

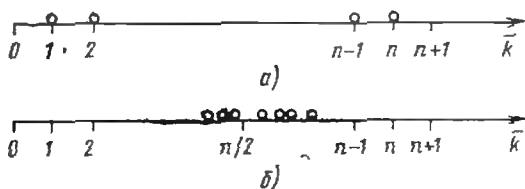


Рис. 3.2.1. Средние ранги на числовой оси: при полностью совпадающих (а), при частично совпадающих (б) мнениях экспертов

ностью расходятся, то средние ранги примерно равны $n/2$. При частично согласованных мнениях ранги группируются вокруг этого среднего значения (см. рис. 3.2.1, б).

Степень рассеяния, как известно, удобно определять с помощью дисперсии. Дисперсия средних рангов по определению равна

$$D(\bar{k}) = \frac{1}{n} \sum_{i=1}^n (\bar{k}_i - M(\bar{k})),$$

где

$$M(\bar{k}) = \frac{1}{n} \sum_{i=1}^n k_i = \frac{n+1}{2}$$

— математическое ожидание среднего ранга.

Определим максимальную дисперсию (она бывает, как легко заметить, при полностью совпадающих мнениях экспертов):

$$D_{\max} = \frac{1}{n} \sum_{i=1}^n \left(i - \frac{n+1}{2} \right)^2 = \frac{n^2 - 1}{12}.$$

Критерий согласованности экспертов удобно представить в виде отношения

$$W = \frac{D(\bar{k})}{D_{\max}} = \frac{12}{n(n^2 - 1)} \sum_{i=1}^n \left(\bar{k}_i - \frac{n+1}{2} \right)^2.$$

Как видно, $0 \leq W \leq 1$. При $W=0$ мнения экспертов полностью расходятся, а при $W=1$ они высказываются единогласно. Таким образом, величина W характеризует степень согласованности экспертов. Чем ближе W к единице, т. е. чем более единодушны эксперты, тем более достоверным можно считать результат ранжирования.

Следует отметить, что эксперты должны высказывать свое мнение независимо друг от друга, т. е. до ранжирования они не должны знать мнения других экспертов. В противном случае возможно появление зависимых мнений, что повышает критерий согласованности W , но не улучшает результатов экспертного опроса, так как зависимым мнением можно просто пренебречь.

Для того чтобы получить представление по поводу конкретного значения критерия согласованности, который никогда не бывает равным ни нулю, ни единице, чтобы считать его малым или большим, можно предложить следующий простой метод. Предположим, что m из N экспертов абсолютно компетентны, а остальные $N - m$ совершенно не компетентны, т. е. принимают свое решение чисто случайно (хотя такого, естественно, не бывает). Тогда дисперсия средних рангов будет образована суммой

$$D(\bar{k}) = \frac{1}{N} [m D_{\max} + (N - m) 0] = \frac{m}{N} D_{\max}$$

Разделив результат на $D_{\max}$, получим $W = m/N$. Таким образом, величина W зависит от числа абсолютно компетентных экспертов (если бы они существовали, разумеется). Так, при $W=0,3$ можно считать, что 30% экспертов были вполне компетентны, а остальные 70% принимали свое решение случайно, что, естественно, могло оказать (а могло и не оказать!) роковое влияние на окончательную ранжировку.

Пронлюстрируем применение этого метода на шуточном примере ранжирования девушек. Рассмотрим довольно обычную ситуацию — разговор трех приятелей о своих девушках. Естественно, что каждый отдает предпочтение своей. Можно ли в данном случае применить метод экспертных оценок для объективного ранжирова-

ния этих девушек по какому-то критерию (если такое ранжирование существует, разумеется)? Оказывается, можно, несмотря на явную пристрастность экспертов — наших приятелей. Пусть x_i — i -я девушка ($i=1, 2, 3$), а $\mathcal{E}_j$ — j -й эксперт ($j=1, 2, 3$). Пусть также априори имеет место следующий объективный ряд ранжирования:

$$x_1 > x_2 > x_3.$$

Здесь знак предпочтения $>$ означает «умнее, чем» или «хозяйственнее, чем» или «красивее, чем» и т. д. в зависимости от выбранного критерия. Но с этим рядом согласен лишь $\mathcal{E}_1$. $\mathcal{E}_2$ считает, что $x_2 > x_1 > x_3$, а $\mathcal{E}_3$ придерживается собственного мнения:

$$x_3 > x_1 > x_2.$$

Как видно, каждый согласен с рядом объективного ранжирования лишь в той его части, которая не касается его девушки. Эти мнения объединяются матрицей типа (3.2.4):

$$\begin{array}{c|ccc} & x_1 & x_2 & x_3 \\ \hline \mathcal{E}_1 & 1 & 2 & 3 \\ \mathcal{E}_2 & 2 & 1 & 3 \\ \mathcal{E}_3 & 2 & 3 & 1 \end{array} \quad \left\| \right\|.$$

Легко убедиться, что средние ранги девушек $\bar{k}_1=5/3$; $\bar{k}_2=2$; $\bar{k}_3=7/3$ отражают объективное ранжирование, т. е. экспертный опрос дал правильный результат.

Определим согласованность наших экспертов:

$$W = D(\bar{k}) / D_{\max} = (2/27) / (2/3) = 1/9,$$

т. е. их мнения оказались очень плохо согласованными, что и следовало ожидать. Тем не менее окончательное ранжирование оказалось правильным. Это произошло за счет осреднения мнений экспертов, которое исключило их индивидуальные особенности, а вместе с ними и их ошибки.

Этот результат показывает эффективность метода экспертных оценок даже в том случае, когда эксперты явно расходятся во мнениях. Важно, чтобы это расхождение касалось различных ранжируемых факторов.

Отсутствие согласованности экспертов может свидетельствовать, с одной стороны, о некомпетентности экспертов, связанной с новизной или слабой изученностью объекта исследования, с другой — о сложности объекта, что затрудняет вынесение решения о рангах факторов. Эксперту проще сопоставить важность некоторых факторов попарно, т. е. указать, ранг какого из двух факторов будет выше. Именно в таких ситуациях обращаются к методу парных сравнений, который мы и рассмотрим ниже.

Метод парных сравнений. Эксперту предлагается про-
ранжировать факторы попарно, т. е. каждой паре фак-
торов x_i и x_l поставить в соответствие число

$$q_{il} = \begin{cases} 1, & \text{если } x_i > x_l, \\ 0, & \text{если } x_i \sim x_l, \\ -1, & \text{если } x_i < x_l. \end{cases}$$

Выражение $x_i > x_l$ означает, что i -й фактор более пред-
почтителен при ранжировании, чем l -й. Знак $\sim$ является
знаком эквивалентности факторов с точки зрения ран-
жирования (или отказа от ранжирования). Числа q_{il}
обладают очевидным свойством $q_{il} = -q_{li}$.

	x_1	$\dots$	x_n
x_1	0	$\dots$	q_{1n}^j
$\vdots$	$\vdots$		$\vdots$
x_n	q_{n1}^j	$\dots$	0

Рис. 3.2.2. Таблица предпочте-
ний при методе парных срав-
нений

	x_1	x_2	x_3
x_1	0	1	-1
x_2	-1	0	1
x_3	1	-1	0

Рис. 3.2.3. Пример противоре-
чивой матрицы

Таким образом, каждый j -й эксперт свое мнение пред-
ставляет в виде матрицы

$$Q^j = \|q_{il}^j\|, \quad i, l = 1, \dots, n; \quad j = 1, \dots, N,$$

где верхний индекс определяет номер эксперта (см. рис.
3.2.2).

Усредним мнения экспертов. Для этого достаточно по-
строить усредненную матрицу размерностью $n \times n$

$$\bar{Q} = \|\bar{q}_{il}\|,$$

где

$$\bar{q}_{il} = \frac{1}{N} \sum_{j=1}^N q_{il}^j$$

— среднее предпочтение i -го фактора l -му. Это и есть
мнение данной группы экспертов.

Определим согласованность этих экспертов. В качестве критерия согласованности аналогично предыдущему естественно выбрать дисперсию величин $\bar{q}_{il}$. В силу того, что их среднее значение равно нулю, получаем

$$D(\bar{q}) = \frac{1}{n(n-1)} \sum_{i,l=1}^n (\bar{q}_{il})^2,$$

где суммирование производится по всей матрице $\bar{Q}$. Максимальное значение дисперсии $D_{\max}=1$ будет иметь место при полной согласованности экспертов. Тогда, вводя критерий согласованности как отношение дисперсии средних предпочтений к максимальной дисперсии, получаем

$$W = \frac{1}{n(n-1)} \sum_{i,l=1}^n (\bar{q}_{il})^2,$$

и в этом случае $0 \leq W \leq 1$, где, очевидно, при $W=1$ мнения экспертов полностью согласованы, а при $W=0$ они противоречат друг другу.

Однако здесь и при полной согласованности эксперты могут противоречить друг другу (пример такой матрицы приведен на рис. 3.2.3, где противоречие имеет вид $x_1 > x_2 > x_3 > x_1$, т. е. оказалось, что $x_1 > x_3$ и $x_3 > x_1$ одновременно). Выявление подобных противоречий совершенно необходимо не только в усредненной матрице, но и во мнении каждого эксперта. Это делается на основе следующего довольно очевидного правила, которое называется *правилом транзитивности*. Для предпочтений оно имеет вид:

$$\text{если } x_1 > x_2 \text{ и } x_2 > x_3, \text{ то } x_1 > x_3; \quad (3.2.5)$$

для эквивалентности

$$\text{если } x_1 \sim x_2 \text{ и } x_2 \sim x_3, \text{ то } x_1 \sim x_3.$$

Матрицы, представляющие мнение каждого эксперта, должны удовлетворять указанному правилу транзитивности. При обнаружении противоречий они возвращаются соответствующему эксперту для исправления замеченных противоречий.

Для определения интересующих нас рангов ранжируемых факторов следует иметь правило вычисления рангов по матрице Q . Таких правил может быть много. Рассмотрим некоторые из них.

Правило 1. Как легко заметить, величина $\bar{q}_{il}$ выражает степень предпочтения i -го фактора l -му. Определим среднее предпочтение каждого фактора всем остальным:

$$\bar{q}_i = \frac{1}{n} \sum_{l=1}^n \bar{q}_{il}, \quad i = 1, \dots, n. \quad (3.2.6)$$

Естественно считать, что первый ранг имеет фактор, среднее предпочтение которого максимально. Так, при

$$\bar{q}_v = \max_{i=1, \dots, n} \{\bar{q}_i\}$$

первый ранг имеет фактор x_v , т. е. $k_v=1$. Аналогично образуются ранги остальных факторов.

Рассмотренное правило, однако, излишне усредняет предпочтения. Так, фактор, имеющий ряд явных (т. е. больших) предпочтений, которые легко обнаруживают эксперты, получит первый ранг только потому, что его второстепенность по отношению к другим факторам была не столь ярко выражена. Именно в этом случае часто приходится обращаться к другому правилу.

Правило 2. Это правило опирается на идею усиления контраста. С этой целью вводится порог δ . Если предпочтение выше этого порога, то оно имеет явный характер, а если ниже, то оно сомнительно, т. е. факторы скорее равноценны. Получается следующее преобразование матрицы средних предпочтений $\bar{Q}$ в контрастную матрицу, элементами которой являются

$$\varphi_{il} = \varphi(q_{il}), \quad i \neq l = 1, \dots, n.$$

где

$$\varphi(q) = \begin{cases} -1, & \text{если } q \leq -\delta, \\ 0, & \text{если } |q| < \delta, \\ 1, & \text{если } q \geq \delta. \end{cases}$$

Как видно, это преобразование целиком и полностью определяется порогом δ ($0 < \delta < 1$). При $\delta=1$ контрастная матрица становится нулевой и все факторы эквивалентны. При $\delta=0$ она полностью заполняется единицами, но при этом неизбежно появление противоречий, т. е. нарушений транзитивности предпочтений (3.2.5). Поэтому при выборе порога δ следует помнить, что его увеличение приводит к отказу от ранжирования, а уменьшение—

к увеличению числа явных предпочтений и к опасности появления противоречий. Одной из возможных рекомендаций по определению оптимального порога является выбор величины δ на «пороге противоречий», т. е. такого значения δ^* , небольшое уменьшение которого приводит к противоречиям.

Пример. Пусть матрица средних предпочтений имеет вид, приведенный на рис. 3.2.4,а. При $\delta=0,7$ контрастная матрица дает следующий ряд ранжирования:

$$x_4 > x_3 > x_2 > x_1. \quad (*)$$

Нули в этой матрице означают не только эквивалентность, но и явно выраженное предпочтение. Поэтому полученное ранжиро-

	x_1	x_2	x_3	x_4	x_1	x_2	x_3	x_4	x_1	x_2	x_3	x_4	x_1	x_2	x_3	x_4
x_1	0	-0,7	-0,44	-0,52	0	-1	0	0	0	-1	0	-1	0	-1	0	-1
x_2	0,7	0	-0,74	0,5	1	0	-1	0	1	0	-1	1	1	0	-1	0
x_3	0,44	0,74	0	-0,78	0	1	0	-1	0	1	0	-1	0	1	0	-1
x_4	0,52	-0,5	0,78	0	0	0	1	0	1	-1	1	0	1	0	1	0
	a)				б)				в)				г)			

Рис. 3.2.4. Матрица средних предпочтений (а); контрастные матрицы при $\delta=0,7$ (б), $\delta=0,5$ (в) и $\delta^*=0,52$ (г)

ние непротиворечиво. В соответствии с алгоритмом понизим порог. При $\delta=0,5$ контрастная матрица (в) уже содержит противоречие, так как $x_4 > x_3 > x_2 > x_4$. Минимальный порог, при котором не получается противоречия, равен $\delta^*=0,52$ (см. рис. 3.2.4,г), что приводит к ранжированному ряду (*), т. е., выбрав порог $\delta=0,52$ и $0,7$, получим тот же ряд (*).

Рассмотрим применение первого правила к этому примеру. Получаем с помощью (3.2.6) из рис. 3.2.4,а $\bar{q}_1=-1,66$; $\bar{q}_2=0,46$; $\bar{q}_3=0,40$; $\bar{q}_4=0,8$. Откуда согласно первому правилу получаем ранжированный ряд вида

$$x_4 > x_2 > x_3 > x_1. \quad (**)$$

Как видно, результат отличается от (*). Это означает, что либо $x_1 \sim x_2$, либо для более точного решения необходимо получить новые данные, которые бы позволили выяснить, какой из двух рядов ранжирования (*) или (**) имеет место в действительности.

3.3. Декомпозиция модели

До сих пор мы рассматривали объект как некий «черный ящик», о внутреннем содержании которого ничего не было известно, и модель представлялась многополюсником с неизвестной структурой.

На следующей стадии структурного синтеза модели естественно использовать дополнительную информацию об объекте как о системе взаимодействующих элементов. Например, такой объект управления, как завод, представляет собой систему взаимодействующих цехов, технологический процесс — систему взаимодействующих операций, организм — систему взаимодействующих органов, а коллектив — систему взаимодействующих членов этого коллектива и т. д.

Информация о подобных структурных особенностях объекта и является исходной для декомпозиции модели объекта, т. е. для «расщепления» модели на взаимодействующие элементы, отражающие сложную многоэлементную структуру объекта. Пример такой декомпозиции показан на рис. 3.3.1, где модель с тремя входами и тремя выходами представлена пятью более простыми элементами (1—5).

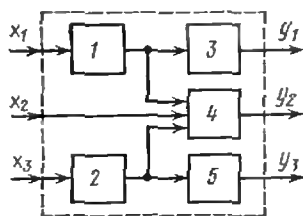


Рис. 3.3.1. Пример декомпозиции модели

Смысл декомпозиции заключается в том, чтобы, воспользовавшись априорными сведениями о структуре объекта, упростить задачу синтеза модели. Это упрощение может идти по двум направлениям. С одной стороны, каждый элемент декомпозированной системы проще исходной (недекомпозированной) системы, что облегчает синтез модели этого элемента и, в конечном счете, модели всего объекта. С другой стороны, упростить задачу синтеза можно независимо от сложности элементов, уменьшив число входов — выходов каждого элемента по сравнению с исходным объектом.

Поясним последнее. Введем численную меру сложности модели как «черного ящика». В основу этой меры естественно положить трудоемкость синтеза модели, т. е. затраты, которые необходимы для создания модели. Будем называть эту меру сложностью. Сложность на ста-

дин анализа (стадии «черного ящика») должна учитывать лишь число входов n и выходов m модели (для простоты не будем различать управляемые и неуправляемые входы). Пусть она имеет вид

$$L=L(n, m). \quad (3.3.1)$$

Относительно вида этой функции можно сформулировать следующие требования. Во-первых, она должна монотонно возрастать по n и m . Во-вторых, число входов n , как правило, сильнее влияет на сложность, чем число выходов. И последнее: эту функцию можно считать аддитивной, т. е. если объект состоит из нескольких g подсистем, то ее сложность равна сумме сложностей этих подсистем, т. е.

$$L=\sum_{i=1}^g L_i, \quad (3.3.2)$$

где L_i — сложность i -й подсистемы исходного объекта. (Заметим, что при большом g это утверждение, строго говоря, неверно, так как сложная система не есть сумма своих частей. Представление (3.3.2) не учитывает трудностей, связанных с разделением объекта на его части.) Всем этим требованиям удовлетворяет, например, следующее выражение (хотя возможны и другие представления):

$$L=n^{\gamma}m, \quad (3.3.3)$$

где $\gamma > 1$, так как число входов сильнее влияет на сложность, чем число выходов. Величину γ следует определять в зависимости от того, во сколько раз увеличивается трудоемкость синтеза модели при увеличении числа ее входов на единицу. Пусть трудоемкость увеличилась в p раз, тогда из (3.3.3) имеем

$$(n+1)^{\gamma}m=pn^{\gamma}m,$$

откуда легко получить выражение для показателя γ :

$$\gamma=\lg p/\lg (n+1/n). \quad (3.3.4)$$

Применяя эту формулу для примера декомпозиции модели, показанного на рис. 3.3.1, получаем для исходной системы

$$L=3^1 \cdot 3=3^{1+1},$$

а для декомпозированной

$$L_d = \sum_{i=1}^5 L_i = 1 + 1 + 1 + 3^1 + 1 = 4 + 3^1.$$

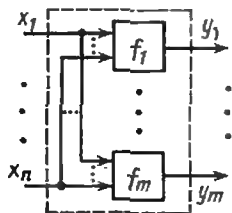
Как видно, $L > L_d$ даже при $\gamma=1$, когда $L=9$, а $L_d=7$. При неизвестном γ его можно считать равным единице, что дает явно заниженную оценку сложности L .

Теперь процесс декомпозиции модели можно рассматривать как процесс минимизации ее сложности, т. е. как решение следующей минимизационной задачи:

$$L_d \rightarrow \min_{D \in \{D\}} \Rightarrow D^*, \quad (3.3.5)$$

где D — операция декомпозиции; $\{D\}$ — множество допустимых данным объектом декомпозиций; D^* — оптимальная декомпозиция, минимизирующая сложность L декомпозируемой системы.

Рис. 3.3.2. Декомпозиция $(n \times m)$ -полосника без учета априорных сведений о структуре объекта



Применим формулу (3.3.3) к декомпозиции, не опирающейся на априорные сведения о структуре объекта (она показана на рис. 3.3.2). Получаем

$$L_d = \sum_{i=1}^m L_i = \sum_{i=1}^m n^1 \cdot 1 = n^1 m,$$

т. е. сложность декомпозированной системы не уменьшилась по сравнению с исходной L . Этого следовало ожидать, так как такая декомпозиция отражает лишь возможность представления векторной модели $\mathbf{Y} = \mathbf{F}(\mathbf{X})$ в скалярной форме

$$y_i = f_i(\mathbf{X}), \quad i=1, \dots, m, \quad (3.3.6)$$

что, естественно, не упрощает задачу. Однако, если из априорных сведений об объекте известно, что некоторые выходы y_i в (3.3.6) не зависят от некоторых входов, то

учет этой информации в процессе такой декомпозиции, как легко заметить, уменьшает сложность декомпозированной системы, т. е. $L_d < L$. Действительно в этом случае $L_d = \sum_{i=1}^m n_i^*$, т. е. $L_d < L$, где n_i — число входов i -го элемента декомпозированной модели ($n_i \leq n$).

Приведем пример декомпозиции объекта на два различных элемента N различными способами (рис. 3.3.3). Это означает, что множество $\{D\}$ состоит из N вариантов, причем для различных вариантов декомпозиции числа k и q принимают различные значения. Очевидно, что хорошей декомпозицией следует считать ту, в которой эти

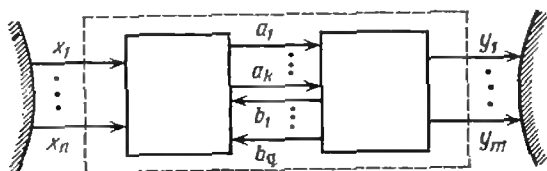


Рис. 3.3.3. Пример декомпозиции на два элемента

числа минимальны. Действительно, из (3.3.2) и (3.3.3) получаем для нашей задачи

$$L(k, q) = (n + q)^* k + k^* (m + q) \rightarrow \min_{\{D\}} \Rightarrow D^*.$$

Оптимальной декомпозицией D^* из $\{D\}$ будет та, которая минимизирует $L(k, q)$. Пусть D_i — i -я декомпозиция, которая определяется двумя числами:

$$D_i = \langle k_i, q_i \rangle, \quad i = 1, \dots, N.$$

Решение поставленной задачи получаем в виде $D^* = D_i$, если

$$L(k_i, q_i) = \min_{i=1, \dots, N} \{L(k_i, q_i)\}.$$

Таким образом, цель декомпозиции модели состоит прежде всего в том, чтобы упростить последующий синтез модели объекта путем ее «расщепления» на более простые элементы. Этот процесс обязательно должен производиться с учетом априорной информации о структурных особенностях объекта.

В результате декомпозиции объекта на g элементов задача синтеза модели объекта сводится к g задачам синтеза модели каждого элемента, т. е. к более простым задачам.

Процесс декомпозиции можно (а иногда и целесообразно) применить и к полученным элементам, декомпозируя их на более мелкие составляющие. Однако, как легко заметить, этот процесс целесообразно заканчивать на определенной стадии, когда дальнейшее дробление объекта лишь усложняет задачу. На рис. 3.3.4,а показана зависимость числа элементов и их сложности от уров-

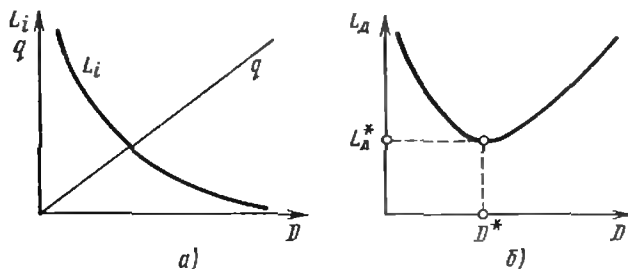


Рис. 3.3.4. Влияние уровня декомпозиции на сложность элементов L_i и их число q (а), а также на сложность декомпозируемого объекта (б)

ня декомпозиции. С увеличением уровня декомпозиции сложность каждого элемента уменьшается не столь интенсивно, что и приводит к увеличению сложности декомпозированного объекта (см. рис. 3.3.4,б). Оптимальная декомпозиция минимизирует сложность декомпозированной системы.

Так или иначе с помощью декомпозиции исходный $(n \times m)$ -полюсник («черный ящик») удастся свести к набору более простых «черных ящичков», чтобы на следующей стадии структурного синтеза определить только их структуру. Сам по себе процесс декомпозиции сводится, таким образом, к определению элементов объекта и установлению взаимосвязи этих элементов в объекте.

3.4. Структура модели

Процесс синтеза оператора F модели объекта F^0 (или ее элементов) сводится к определению структуры St этой модели и ее параметров $C = (c_1, \dots, c_k)$, т. е. опера-

гор модели F в соответствии с (В.4.2) представляется в виде пары

$$F = \langle St, C \rangle, \quad (3.4.1)$$

а связь входов и выходов модели в виде

$$Y = F_{St}(X, U, C), \quad (3.4.2)$$

где F_{St} — оператор преобразования со структурой St , параметры которого для удобства вынесены в переменные C . (Индекс St будем в дальнейшем опускать, так как оператор без структуры не существует.) Представление оператора преобразования в виде (3.4.2) и является результатом параметризации модели, для которой определяется структура St , а неизвестными остаются параметры модели C .

Рассмотрим основные структурные категории: динамичность, нелинейность, стохастичность и нестационарность модели объекта.

Динамичность. С точки зрения динамики объекта управления целесообразно рассмотреть статическую и динамическую структуры объекта.

Структуру объекта будем называть статической, если для его описания достаточно привлечь модель в виде функции F , и динамической, если для адекватного описания необходимо привлечь операторы с памятью (например, дифференциальные операторы, интеграторы, запаздывание и т. д.).

Примером статической структуры является структура, образуемая разложением выхода объекта по определенной системе линейно-независимых функций входов:

$$y = \sum_{i=1}^k c_i \varphi_i(X, U), \quad (3.4.3)$$

где $\{\varphi_i(\cdot, \cdot)\}$ — заданная система функций входов, выбор которой и определяет выбор статической структуры объекта; $c_1, \dots, c_k$ — параметры объекта, определяемые на этапе идентификации (см. гл. 5). Вспомним пример с коктейлем (§ В.5), модель которого представляла собой статическую структуру с функциями вида $\varphi_i(X, U) = u_i$ ($i=1, 2, 3$).

Примером динамической структуры при $n=m=1$ ($X=x, Y=y$) является модель, описываемая линейным

дифференциальным уравнением

$$\frac{d^p y}{dt^p} + a_{p-1} \frac{d^{p-1} y}{dt^{p-1}} + \dots + a_1 \frac{dy}{dt} + a_0 y = b_l \frac{d^l x}{dt^l} + \dots + b_1 \frac{dx}{dt} + b_0 x \quad (3.4.4)$$

или в более компактной форме

$$\sum_{i=1}^p a_i y^{(i)} = \sum_{j=1}^l b_j x^{(j)},$$

где $y^{(i)} = \frac{d^i y}{dt^i}$, $x^{(j)} = \frac{d^j x}{dt^j}$; a_i ($i=0, \dots, p-1$) и b_j ($j=0, \dots, l$), $a_p = 1$ — параметры непрерывной модели.

В дискретном случае уравнение (3.4.3) записывается в рекуррентной (конечно-разностной) форме:

$$y_r = \sum_{i=1}^p d_i y_{r-i} + \sum_{j=1}^{l+1} e_j x_{r-j+1}, \quad (3.4.5)$$

где $d_1, \dots, d_p$ и $e_1, \dots, e_{l+1}$ — параметры дискретной модели, а x_r и y_r — значения входа и выхода модели в r -й момент времени. (Время в этом случае предполагается дискретным, т. е. $t=r=1, 2, \dots$)

Например, модель такого простого объекта, как маятник при малых отклонениях, имеет структуру вида

$$\frac{d^2 y}{dt^2} + a_1 \frac{dy}{dt} + a_0 y = b_0 x(t),$$

где y — угол отклонения маятника от вертикали, а $x(t)$ — внешняя возмущающая сила. В дискретном случае модель маятника записывается в виде конечно-разностного выражения

$$y_r = d_1 y_{r-1} + d_2 y_{r-2} + e_1 x_r.$$

Заметим, что конечно-разностный вид модели более распространен для описания поведения сложных систем. Мы его будем широко применять в дальнейшем.

Нелинейность. С точки зрения реакции на внешнее воздействие объекты удобно подразделять на линейные и нелинейные. Под линейным объектом будем подразумевать такой объект, реакция которого на сумму двух

любых внешних возмущений $X_1(t)$ и $X_2(t)$ равна сумме реакций на эти возмущения, т. е.

$$F^o(X_1(t) + X_2(t)) = F^o(X_1(t)) + F^o(X_2(t)), \quad (3.4.6)$$

причем здесь необходимо еще соблюдение условия $F^o(0) = 0$, т. е. объект должен быть центрирован.

Линейная статическая структура описывается обычной линейной зависимостью

$$y = c_1 x_1 + \dots + c_n x_n$$

и в общем виде (при линейности по X и U)

$$y = \sum_{i=1}^n c_i x_i + \sum_{j=1}^q c_{n+j} u_j. \quad (3.4.7)$$

Примером такой зависимости являются свойства коктейля (вкус, крепость и стоимость), которые линейно зависят от состава (см. § В.5). Легко убедиться, что динамические модели (3.4.4) и (3.4.5) также линейные. Для этого достаточно их подставить в (3.4.6).

Типичным примером нелинейной статической структуры являются вкусовые ощущения. Например, при добавлении соли к несоленой пище вкус ее сначала улучшается, а затем при дальнейшем добавлении резко падает (этот тот самый «пересол», который «на спине»). (Следует отметить, что в примере о коктейле, рассмотренном в § В.5, вкусовая зависимость $f_1(U)$ для простоты выбрана линейной, что является лишь приближением, необходимым для наглядного решения задачи до конца. В действительности эта функция нелинейна).

Маятник при больших отклонениях является другим примером нелинейной структуры вида

$$\frac{d^2 y}{dt^2} + a_1 \frac{dy}{dt} + a_0 \sin y = b_0 x(t).$$

Стохастичность. Стохастический характер структуры объекта связан с наличием в объекте и в среде различных неконтролируемых, но существенных факторов, которые можно моделировать статистически:

$$Y = F(X, U, E(t)), \quad (3.4.8)$$

где $E(t)$ — случайный процесс, моделирующий имеющуюся неопределенность объекта и среды. Эта неопределенность может быть связана как с быстрым измене-

нием параметров объекта, так и с помехами, накладываемыми на измерения входа и выхода объекта.

Так или иначе, но стохастический объект ведет себя неоднозначно в одинаковых ситуациях, что моделируется случайным вектором $E(t)$, статистические свойства которого должны быть заданы. В простейшем случае имеем:

$$Y = F(X, U) + E(t). \quad (3.4.9)$$

Примером такого стохастического объекта является любой биологический организм, который, как известно, в одинаковых условиях ведет себя по-разному. В этом случае выражение $F(X, U)$ описывает поведение объекта, которое строго зависит от внешних условий, а все отклонения от этого регулярного поведения образуют «случайную помеху» $E(t)$.

Нестационарность. Нестационарность объекта связана с детерминированным или случайным изменением во времени оператора F^0 объекта. Если это изменение происходит достаточно медленно, в виде дрейфа параметров, то его можно не учитывать, так как процесс коррекции (адаптации) модели на каждом шаге управления позволяет подстраивать модель и тем самым компенсировать указанный дрейф. При быстром изменении F^0 характер нестационарности необходимо учесть в структуре модели в виде зависимости F от времени:

$$Y = F_t(X, U, C) = F(X, U, C, t),$$

где время t является параметром.

Например, при линейном изменении параметров модели получаем

$$C = C_0 + C_1 t,$$

где C_0, C_1 — параметры модели и ее изменения, которые должны быть определены на последующем этапе идентификации.

Примером такой нестационарности является «старение» объекта, т. е. его амортизация во времени. Это старение выражается в определенном изменении его характера, что и делает объект нестационарным.

Следует отметить, что указанные структурные категории еще не определяют структуры модели, а скорее тип ее структуры. Сам по себе процесс выбора структуры модели объекта или его элементов сводится, по сути

дела, к выбору языка описания объекта. Это описание и есть модель.

Примерами такого рода языков являются линейная алгебра—для синтеза линейных статистических моделей типа (3.4.7), линейные дифференциальные уравнения—для синтеза непрерывных линейных динамических моделей типа (3.4.4), конечно-разностные уравнения—для синтеза дискретных линейных динамических моделей типа (3.4.5) и т. д. Однако эти языки классической математики пригодны для описания сравнительно несложных объектов. Сложные объекты управления требуют создания новых языков с большими выразительными средствами. Сейчас существует уже много таких языков.

Рассмотрим два направления в этой области—имитационные и семиотические подходы к моделированию сложных объектов.

3.5. Имитационные модели

Под имитационным моделированием будем понимать пошаговое моделирование поведения объекта с помощью ЭВМ. Это означает, что фиксируются определенные моменты времени $t_1, \dots, t_N$ и состояние модели определяется (вычисляется ЭВМ) последовательно в каждый из этих моментов времени. Для этого необходимо задать правило (алгоритм) перехода модели из одного состояния в следующее, т. е. преобразование $Y_i \rightarrow Y_{i+1}$, где Y_i —состояние модели в i -й момент времени. Пусть, как обычно, состояние модели определяется вектором $Y = (y_1, \dots, y_m)$, т. е. m числами, состояние среды $X = (x_1, \dots, x_n)$ — n числами, а состояние управления $U = (u_1, \dots, u_q)$ — q числами. Тогда имитационная модель определяется заданием оператора F , с помощью которого можно определить состояние модели в последующий момент времени (Y_{i+1}), зная состояние в предыдущий Y_i и значения X_{i+1} и U_{i+1} :

$$(Y_i, X_{i+1}, U_{i+1}) \xrightarrow{F} Y_{i+1}. \quad (3.5.1)$$

Грубо говоря, такая модель позволяет определить, что будет завтра, если известно, что имеется сегодня. Специфика имитационного моделирования и состоит в том, что состояние объекта определяется рекуррентно на каждом шаге, исходя только из предыдущего шага. Преобразо-

вание (3.5.1) удобно записать в виде рекуррентной формулы

$$Y_{i+1} = F(Y_i, X_{i+1}, U_{i+1}), \quad (3.5.2)$$

где F — оператор имитаций изменения состояния модели. Этот оператор и определяет имитационную модель объекта управления с ее структурой St и параметрами C в соответствии с (3.4.1).

На стадии структурного синтеза имитационной модели F определяется ее структура St и содержательный смысл параметров $C = (c_1, \dots, c_k)$, численное значение которых выясняется на этапе идентификации (см. следующую главу).

Как легко заметить, имитационные модели обобщают конечно-разностную модель, рассмотренную в § 3.4. Рассмотрим, например, нелинейную конечно-разностную модель, записанную в виде

$$y_r = f(y_{r-1}, \dots, y_{r-p}, x_r, \dots, x_{r-s}),$$

где f — заданная функция. Введем обозначения

$$Y_{r-1} = (y_{r-1}, \dots, y_{r-p}); \quad X_r = (x_r, \dots, x_{r-s}).$$

Тогда уравнение для y_r запишется в виде

$$y_r = f(Y_{r-1}, X_r). \quad (3.5.3)$$

Как видно, это выражение представляет собой частный случай (3.5.2). Следовательно, конечно-разностные модели являются частным случаем имитационной модели, так же как любая функция является частным случаем алгоритма (ведь функция тоже является правилом вычисления ее значения по аргументу, т. е. алгоритмом).

Имитационные модели обладают еще одним достоинством — они позволяют учитывать неконтролируемые факторы E объекта, его стохастичность (3.4.8) в виде

$$Y_{i+1} = F(Y_i, X_{i+1}, U_{i+1}, E_{i+1}), \quad i = 1, \dots, N, \quad (3.5.4)$$

где, однако, необходимо знать, каким образом фактор E влияет на состояние Y объекта. Для этого следует хорошо разобраться в объекте и указать совершенно точно, как входит неконтролируемый фактор E в оператор объекта с тем, чтобы эти данные отразить в операторе F модели. Для работы с такой моделью необходимо знать конкретные значения фактора E , который, как известно, наблюдаем.

Это противоречие позволяет разрешить так называемый метод Монте-Карло (или метод статистических испытаний). Для его реализации необходимо знать некоторые статистические свойства фактора E (например, закон его распределения, корреляционные свойства и т. д.). Эти свойства, вообще говоря, могут зависеть от Y , X и U . Располагая этими сведениями, можно моделировать ненаблюдаемый фактор в виде случайных рядов

$$E^j_1, E^j_2, \dots, E^j_N, \quad j=1, \dots, N', \quad (3.5.5)$$

где нижний индекс соответствует моменту дискретного времени, а верхний — номеру моделируемого ряда (всего моделируется N' таких статистически эквивалентных рядов). Разумеется, что ни один из этих рядов не является точной реализацией действительности, но каждый имеет такие же статистические свойства, что и реальный. Именно поэтому ряды (3.5.5) позволяют исследовать статистические свойства модели (3.5.4). Например, поведение модели «в среднем» описывается следующим образом:

$$\bar{Y}_i = \frac{1}{N} \sum_{j=1}^{N'} Y^j_i, \quad i=1, \dots, N,$$

где Y^j_i — j -я реализация поведения модели в i -й момент времени:

$$Y^j_i = F(Y^j_{i-1}, X_i, U_i, E^j_i), \quad i=1, \dots, N-1.$$

Дисперсия выхода модели вычисляется очевидным образом:

$$D(Y_i) = \frac{1}{N'-1} \sum_{j=1}^{N'} (Y^j_i - \bar{Y}_i)^2.$$

Аналогично можно оценить и другие статистические характеристики имитируемого процесса.

В этом и заключается метод Монте-Карло — он позволяет оценить статистические свойства поведения объекта путем вероятностного «разыгрывания» поведения модели, причем одна реализация поведения отличается от другой различными значениями ненаблюдаемого фактора E .

3.6. Семиотические модели

Синтез всякой модели является процессом формализации описания объекта, т. е. переходом от естественного языка к формализованному (графу, имитационной модели, аналитическим уравнениям и т. д.), что всегда представляет трудности, особенно для сложных объектов управления. До сих пор эта операция является в значительной мере искусством. Это связано, прежде всего, с многозначностью и «размытостью» понятий естественного языка. Однако при описании объектов управления с этой трудностью можно справиться, если заранее оговорить смысл употребляемых понятий, т. е. составить нечто вроде толкового словаря.

В этом случае описание на естественном языке можно формализовать, т. е. прямо записать в символической форме с помощью небольшого числа определенных знаков — понятий, связанных непосредственно с понятиями языка. Назовем такую модель *семиотической* (семиотика — наука о знаковых системах), так как она имеет ярко выраженный знаковый характер и поэтому очень близка к естественному языку. Преимущества семиотических моделей очевидны — с их помощью можно описывать очень широкий класс объектов управления — практически все объекты, которые могут быть описаны на естественном языке.

Основой построения семиотической модели является язык описания этих моделей. Рассмотрим один из этих языков — язык бинарных отношений. Он достаточно прост.

Словарем языка бинарных отношений являются:

— базовые понятия (будем обозначать их символами a_i , $i=1, 2, \dots$), которые порождаются объектом и вводятся на основе его описания. Грубо говоря, это элементы, из которых состоит описываемый объект, например a_1 — «лифт», a_2 — «кнопка» и т. д.

— базовые отношения (будем обозначать их символами r_j , $j=1, 2, \dots$), которые указывают на отношения между базовыми понятиями. Необходимость такого рода отношений очевидна, так как для описания объекта мало описать его элементы, нужно еще указать, в каком отношении эти элементы находятся друг к другу. Эти отношения универсальны; для естественного языка выделено около 200 отношений, например r_1 — «находиться в», r_2 — «быть одновременно» и т. д. Заметим, что

такого рода отношения и образуют специфику этого языка, который постулирует их бинарность, т. е. данное отношение между двумя понятиями или есть или нет, а третьего не дано;

— имена, т. е. различные названия предметов, которые используются в естественном языке (будем их обозначать буквами b_k , $k=1, 2, \dots$). Для введения имен имеется специальное базовое отношение r_0 — «иметь имя». Имена кроме обычных имен собственных типа «Саша», «Маша» могут отражать важную специфику и конкретизацию понятий, например b_1 — «пассажирский», b_2 — «грузовой», $b_3=5$ (число «5» — это имя) и т. д.

Таков словарь языка бинарных отношений. Он очень удобен и конструктивен, каждое его слово имеет конкретное содержание, связанное с описываемым объектом управления. Действительно, понятия и имена в него вводятся по мере необходимости, и каждый объект порождает свой индивидуальный словарь базовых понятий, отношений и имен. Это семантика языка бинарных отношений.

Теперь о грамматике (точнее, синтаксисе) этого языка. Синтаксис должен точно и недвусмысленно указывать, каким образом строятся синтаксически правильные фразы, т. е. определять формальные признаки синтаксически правильных фраз (они могут быть ошибочны, как, например, «дом является частью комнаты», или бессмысленны, например, «проводка одновременно с лифтом», тем не менее они синтаксически правильны).

Синтаксически правильные фразы (A) имеют вид:

$A=a_i$, т. е. любое базовое понятие является синтаксически правильной фразой;

$A=(a_i r_j a_k)$, где a_i и a_k — любые базовые понятия; r_j — любое базовое отношение, кроме r_0 («иметь имя»), т. е. два базовых понятия, объединенные базовым отношением, также являются синтаксически правильной фразой;

$A=(A_i r_j A_k)$, где A_i и A_k — синтаксически правильные фразы, а r_j — любое базовое отношение, кроме r_0 , т. е. две синтаксически правильные фразы, объединенные базовым отношением, являются синтаксически правильной фразой;

$A=(A_j r_0 b_k)$, где A_j — синтаксически правильная фраза; b_k — любое имя, т. е. выражение « A_j имеет имя b_k » всегда является синтаксически правильной фразой.

Других видов синтаксически правильных фраз нет. Последнее замечание исключает возможность появления каких-то других фраз. Для ЭВМ это очень важно, так как она не обладает опытом человека, позволяющим сразу огсенвать неправильные конструкции.

Пример. Пусть объектом нашего анализа будет ситуация, описанная фразой естественного языка «Саша и Маша находятся в пассажирском лифте». Эта ситуация может быть описана на языке бинарных отношений следующим образом. Словарь:

$$\begin{aligned} \text{Базовые понятия:} & \begin{cases} a_1 — \text{„человек“}, \\ a_2 — \text{„лифт“}. \end{cases} \\ \text{Базовые отношения:} & \begin{cases} r_0 — \text{„иметь имя“}, \\ r_1 — \text{„находиться в“}, \\ r_2 — \text{„быть одновременно“}, \end{cases} \\ \text{Имена:} & \begin{cases} b_1 — \text{„Саша“}, \\ b_2 — \text{„Маша“}, \\ b_3 — \text{„пассажирский“}. \end{cases} \end{aligned}$$

Начнем с простейших фраз, которые войдут в описание ситуации:

$$\begin{aligned} A_1 &= (a_1 r_0 b_1) — \text{„человек по имени Саша“}, \\ A_2 &= (a_1 r_0 b_2) — \text{„человек по имени Маша“}, \\ A_3 &= (a_2 r_0 b_3) — \text{„пассажирский лифт“}, \\ A_4 &= (A_1 r_2 A_2) — \text{„человек Саша и одновременно человек Маша“} \end{aligned}$$

Теперь описываемая ситуация естественно представляется в виде: $A_5 = (A_4 r_1 A_3)$ — «человек Саша и человек Маша одновременно находятся в пассажирском лифте».

Добавление понятия «человек» здесь не является бюрократизмом. Оно необходимо, так как имена могут иметь не только люди. Например, прибор МАША — это какой-нибудь Модифицированный Аппарат с Широкополосным Атенюатором. Тогда та же исходная ситуация («Саша и Маша находятся в лифте») запишется иначе. Для этого в словарь следует ввести новое базовое понятие a_3 — «прибор». Образует пужные фразы:

$$\begin{aligned} A_6 &= (a_3 r_0 b_2) — \text{„прибор с названием МАША“}, \\ A_7 &= (A_1 r_2 A_6) — \text{„человек Саша одновременно с прибором МАША“}. \end{aligned}$$

В результате получаем $A_8 = (A_7 r_1 A_3)$ — «человек Саша и прибор МАША одновременно находятся в пассажирском лифте».

Вводя дополнительные скобки, можно описанные ситуации записать символами словаря:

$$\begin{aligned} A_5 &= (((a_1 r_0 b_1) r_2 (a_1 r_0 b_2)) r_1 (a_2 r_0 b_3)), \\ A_8 &= (((a_1 r_0 b_1) r_2 (a_3 r_0 b_2)) r_1 (a_2 r_0 b_3)). \end{aligned}$$

Это осуществляется прямой подстановкой выражений для A_1 и A_2 в A_4 , которое вместе с A_3 подставляется в A_5 и т. д. Естественно, что мало кого волнует судьба Саши и Маши в лифте — это только наглядный пример описания некоей сложившейся ситуации.

Описанная ситуация статическая. Она интересует нас до тех пор, пока важен факт пребывания Саши и Маши в лифте. Если же нас интересует процесс функционирования лифта, то необходимо ввести динамику, т. е. изменение ситуации во времени. Для того чтобы описать движение лифта, необходимо ввести дополнительное базовое понятие a_4 — «этаж», имена для него: 1, 2, 3, ..., N и базовое отношение r_3 — «находиться на». Тогда движение лифта вверх будет сопровождаться последовательностью ситуаций:

$$B_1 = (a_2 r_3 (a_4 r_0 1)) \text{ — «лифт находится на первом этаже»}$$

$$B_N = (a_2 r_3 (a_4 r_0 N)) \text{ — «лифт находится на } N\text{-м этаже»}.$$

Если имеются какие-то внешние воздействия на объект, зависящие от времени, то необходимо ввести базовое понятие a_5 — «время» и имена этого времени: 1, 2, 3, ... (в удобных единицах — минутах, часах и т. д.), которые изменяются от одной ситуации к другой.

Кроме того, необходимо ввести базовое отношение r_4 — «порождает», которое позволяет вводить внешние воздействия с заданными именами $g_1, \dots, g_L$ как «порождение» определенных моментов времени: $(a_5 r_0 i) r_4 (a_j r_0 g_i)$, т. е. «в i -й момент времени a_j приобретает имя g_i ».

Например, с помощью базовых понятий d_j — « j -я кнопка в лифте» ($j=1, \dots, N$) и двух имен ($L=2$): g_1 — «нажата», g_2 — «не нажата» можно воздействовать на программу движения лифта.

Управление связано с образованием определенных ситуаций в объекте, оно сводится к изменению имен базовых понятий, используемых для описания объекта. Синтез управления состоит в образовании фраз

$$(B_j r_4 (a_i r_0 g_k)), \quad j=1, \dots, \quad (3.6.1)$$

каждая из которых указывает, какое имя (g_k) приобретает понятие a_i в ситуации B_j . Например, если

$B_j = (a_2 r_3 (a_4 r_0 j))$ — «лифт находится на j -м этаже»,
то выражения

$(B_N r_4 (d_1 r_0 g_1))$ — «на N -м этаже нажимается первая кнопка»,
 $(B_1 r_4 (d_N r_0 g_1))$ — «на первом этаже нажимается N -я кнопка»

означают, что наши «Саша и Маша катаются на лифте». Здесь еще необходимо добавить условие

$(B_j r_4 (d_j r_0 g_2))$ — «на j -м этаже j -я кнопка выключается»,
выполнение которого возлагается на автоматику лифта.

Как видно, язык бинарных отношений позволяет описывать символически весьма сложные ситуации в сложных системах. Однако само по себе символическое описание почти ничего не дает (кроме, разве, сокращения записи, что иногда бывает немаловажным обстоятельством). Символическое описание объекта должно обеспечивать возможность выявления каких-то свойств этого объекта, которые неочевидно следуют из его описания. Для этого используются разного рода формальные преобразования, с помощью которых можно изменять в нужную сторону исходное описание объекта. Причем

делать это может ЭВМ, так как эти преобразования формализованы.

Преобразования опираются на алгебраические свойства базовых отношений r_i . Так, упомянутое выше отношение r_2 («быть одновременно») обладает свойством симметрии. Это означает, что $(A_1 r_2 A_2) = (A_2 r_2 A_1)$, т. е., если A_1 происходит одновременно с A_2 , то и A_2 происходит одновременно с A_1 . Отношение r_1 («находится в») транзитивно, т. е. из $(A_1 r_1 A_2)$ и $(A_2 r_1 A_3)$ следует $(A_1 r_1 A_3)$. Действительно, если A_1 находится внутри A_2 , а A_2 — внутри A_3 , то A_1 находится внутри A_3 (как в известной сказке: игла — в яйце, яйцо — в утке, утка — в сундуке и т. д.; следовательно, игла в сундуке — это и есть сказочное использование алгебраического свойства транзитивности).

В настоящее время выявлено почти 400 таких свойств, с помощью которых можно преобразовать исходное описание.

Другим способом преобразования такого описания объекта является образование новых понятий. Под новым понятием здесь будем подразумевать ситуацию, часто повторяющуюся в описании объекта и связанную с целями управлением. Так, «такси, движущееся без пассажира» — ситуация, которая очень волнует таксистов (именно поэтому возникло жаргонное понятие «холостяк»). Обобщаются ситуации, которые отличаются лишь базовыми понятиями (обобщение по базовым понятиям), базовыми отношениями (обобщение по базовым отношениям) и именами (обобщение по именам). Например, при описании работы почты очень важно определить постоянных получателей. Ими могут быть отдельные люди, учреждения, заводы, магазины и т. д. Все они являются потребителями корреспонденции. Естественнo обобщить их одним понятием «абонент».

При описании многочисленных свойств какого-то объекта, связанных с его отношением к измерительным приборам, естественно эти отношения объединить и обобщить, образовав понятия химических, физических, социальных свойств и т. д.

Наиболее распространенно обобщение по имени. Важно ли с точки зрения управления лифтом, что имена его пассажиров Саша и Маша? Скорее всего нет. Поэтому описание работы лифта естественно обобщить по именам пассажиров. Это вместо громоздкого описания

«человек Саша и человек Маша одновременно находятся в лифте» воспользоваться столь же информативным «Два человека в лифте».

Описанные способы преобразования исходного громоздкого описания объекта позволяют получить его в более компактном виде и приступить на следующем этапе к идентификации неизвестных параметров описания (которые обычно содержат имена) и синтезу управления (3.6.1), т. е. к определению таких имен и моментов их введения, которые позволяют на этой модели достигнуть целей управления.

ЛИТЕРАТУРНЫЙ КОММЕНТАРИЙ К ГЛАВЕ 3

Задача использования формальных подходов при определении структуры модели является достаточно новой. Поскольку априорная информация о структуре объекта управления очень трудно формализуема, здесь пока почти полностью господствуют экспертные методы. Систематических руководств по структурному синтезу достаточно широкого класса моделей объектов управления пока нет. Можно лишь привести одну главу книги [1].

Комментарий к методу экспертных оценок был уже сделан в гл. 2 с указанием рекомендуемой литературы. Имитационные модели, рассмотренные в § 3.5, являются сейчас самым распространенным классом моделей, используемых для описания поведения сложных систем управления. К числу их достоинств следует отнести сочетаемость с классическими (аналитическими) моделями и моделями Монте-Карло, использующими идею статистических испытаний. В настоящее время разработаны эффективные и тонкие методы имитационного моделирования, позволяющие синтезировать модели сложных технических, технологических [3, 4] и даже социально-экономических [12, 13] систем. Эти модели позволяют управлять сложными объектами, опираясь в основном на прогноз поведения такого рода объектов с помощью имитации.

Для первого знакомства с имитационными моделями можно порекомендовать [6—8, 15]. С основными методами Монте-Карло можно ознакомиться по [9] и более детально по книгам [2, 10, 14]. К книгам [2—5 и 12, 13] следует обратиться при более тщательном изучении имитационного моделирования.

В последнее время появился новый класс моделей — семиотические, рассмотренные в § 3.6. Они позволяют формализовать объекты, поведение которых описывается на языке, близком к естественному. По своим возможностям семиотические модели уступают только естественному языку. Это обстоятельство позволяет их считать наиболее перспективными моделями описания сложных объектов управления. С основными чертами и применениями семиотических моделей можно ознакомиться по брошюре [11], а язык бинарных отношений изложен подробно в [16], правда для информационно-поисковых систем.

СПИСОК ЛИТЕРАТУРЫ

1. Растрингин Л. А., Маджаров Н. Е. Введение в идентификацию объектов управления. — М.: Энергия, 1977, с. 33—49.

2. Ермаков С. М. Метод Монте-Карло и смежные вопросы. — М.: Наука, 1971, с. 86—104.
3. Бусленко Н. П. Математическое моделирование производственных процессов на ЦЭВМ. — М.: Наука, 1964.
4. Бусленко Н. П. Моделирование сложных систем. — М.: Наука, 1968.
5. Вагнер Г. Основы исследования операций: Пер. с англ. — М.: Мир, 1973. — Т. 3, с. 343—392.
6. Мойсеев Н. Н. Математик задает вопросы. — М.: Знание, 1974, с. 122—146.
7. Мойсеев Н. Н. Имитационные модели. — В кн.: Наука и человечество: Международный ежегодник. — М.: Знание, 1973.
8. Корявов П. П., Сушков Б. Г. Имитация динамических процессов. — М.: Знание, 1973.
9. Соболев И. М. Метод Монте-Карло. — М.: Наука, 1968.
10. Полляк Ю. Г. Вероятностное моделирование на ЭВМ. — М.: Сов. радио, 1971.
11. Поспелов Д. А. Большие системы: Ситуационное управление. — М.: Знание, 1975.
12. Нейлор Т. Машинные имитационные эксперименты с моделями экономических систем: Пер. с англ. — М.: Мир, 1975.
13. Форрестер Дж. Мировая динамика: Пер. с англ. — М.: Наука, 1978. Мир, 1974.
14. Метод статистических испытаний. — М.: Наука, 1962.
15. Шеннон К. Имитационное моделирование: Пер. с англ. — М.: Наука, 1979.
16. Информационно-поисковая система «БИТ». — Киев: Наукова думка, 1968.

ГЛАВА 4

ИДЕНТИФИКАЦИЯ ПАРАМЕТРОВ МОДЕЛИ

Теперь, когда структура St оператора F модели выяснена, необходимо определить неизвестные параметры $c_1, \dots, c_k$ этой модели. Как сказано выше (во введении), эту операцию в зависимости от вида оператора F можно реализовать двумя путями: идентификацией и планированием экспериментов с объектом. Первый подход (идентификация) позволяет определить параметры связи входа и выхода объекта (параметры модели) в случае, если есть возможность наблюдения за изменяющимся входом объекта, т. е. в режиме его нормальной эксплуатации. Второй (планирование эксперимента) применяют в случае, когда управляемый вход объекта не изменяется во времени и необходимо спланировать специальные эксперименты для определения параметров модели, связывающей этот вход с выходом объекта. В этой главе мы рассмотрим идентификацию параметров модели.

4.1. Постановка задачи идентификации

Итак, под идентификацией модели мы подразумеваем процесс определения ее параметров

$$C = \{c_1, \dots, c_n\} \quad (4.1.1)$$

в режиме нормальной эксплуатации объекта. Структура модели при этом известна (она определена на стадии структурного синтеза):

$$Y = F(X, U, C), \quad (4.1.2)$$

т. е. оператор F предполагается заданным. Это означает, что задан алгоритм (правило, инструкция), с помощью которого можно определить состояние Y модели, если заданы состояния X и U ее входов, а также параметры (4.1.1). Именно эти параметры определяются на этапе идентификации.

Очевидно, что для идентификации необходимо иметь информацию об изменении входов и выходов объекта. Но объект пока не управляется (мы только создаем систему управления), поэтому влияние входа U на выход Y не может быть исследовано на этапе идентификации. Это несколько упрощает задачу, так как вместо модели (4.1.2) следует брать модель вида

$$Y = F'(X, C), \quad (4.1.3)$$

в которой не фигурирует управляемый вход U (его влияние на выход будет исследовано на этапе планирования эксперимента).

В процессе идентификации используются исходные данные, которые удобно подразделить на два класса:

— *априорные*, которые содержатся в структуре S_I модели. Это означает, что должен быть задан (или определен на этапе структурного синтеза, рассмотренного в гл. 3) вид оператора F' . Например, вид уравнения, граф взаимосвязи элементов модели и т. д.;

— *апостериорные*, которые представляют собой наблюдения состояний входа X и выхода Y объекта в процессе его нормальной эксплуатации, т. е. информацию

$$I = \langle X_i, Y_i \rangle, \quad i = 1, \dots, N, \quad (4.1.4)$$

где i — номер моментов времени t_i , когда фиксировались значения X и Y , т. е. $X_i = X(t_i)$, $Y_i = Y(t_i)$, где $X(t)$ и $Y(t)$ — функции, описывающие поведение входа и вы-

хода объекта в процессе его нормального функционирования в среде. Моменты времени t_i обычно равномерно покрывают промежуток времени наблюдения $[0, T]$, т. е. $t_i = \tau(i-1)$, где τ — интервал между наблюдениями (4.1.4), т. е. $\tau = T/(N-1)$.

Таким образом, исходные данные, необходимые для идентификации, образуются двойкой

$$\langle St, I \rangle, \quad (4.1.5)$$

т. е. структурой модели (4.1.3) и наблюдениями (4.1.4). Процесс идентификации параметров модели сводится к определению параметров (4.1.1) по исходным данным (4.1.5), т. е.

$$C = \varphi(St, I), \quad (4.1.6)$$

где φ — алгоритм идентификации, определяющий, каким образом можно найти параметры C , зная St и I .

В этой главе будут рассмотрены различные алгоритмы φ . Эти алгоритмы подразделяются на два больших класса: адаптивные и неадаптивные.

Под *адаптивным алгоритмом идентификации* понимают алгоритм, позволяющий уточнять значения идентифицируемых параметров модели по мере получения дополнительной информации о работе объекта. Пусть на i -м шаге адаптивной идентификации были какие-то определенные значения идентифицируемых параметров. Отметим их индексом i :

$$C_i = (c_{i1}, \dots, c_{in}).$$

Пусть, далее, получена дополнительная информация, т. е. пара наблюдений входа и выхода объекта в $(i+1)$ -й момент времени:

$$I_{i+1} = \langle X_{i+1}, Y_{i+1} \rangle. \quad (4.1.7)$$

Очевидно, что эта информация должна каким-то образом изменить (откорректировать) имеющиеся значения C и дать возможность получить C_{i+1} — более точное значение параметров. Связь между C_i и C_{i+1} определяется адаптивным алгоритмом идентификации:

$$(C_i, I_{i+1}) \xrightarrow{\tilde{\varphi}_a} C_{i+1}$$

или в обычной рекуррентной форме

$$C_{i+1} = \tilde{\varphi}_a(C_i, I_{i+1}). \quad (4.1.8)$$

Здесь $\tilde{\varphi}_a$ — алгоритм адаптивной идентификации, который позволяет определить последующее значение параметров, исходя из новой информации (I_{i+1}) и старых представлений о значениях параметров C_i . Адаптация, таким образом, представляет собой способ получения «нового знания» путем коррекции «старого знания» на основе новой информации.

Алгоритм (4.1.8) удобнее записать в виде

$$C_{i+1} = C_i + \varphi_a(F(X_{i+1}, C_i), I_{i+1}), \quad (4.1.9)$$

где φ_a — оператор адаптивной идентификации.

Если адаптивный метод идентификации реализуется в реальном масштабе времени, то его называют *методом самонастраивающейся модели*. Схема этого метода показана на рис. 4.1.1. Здесь на вход модели подается вход X объекта. Информация о состоянии объекта Y , модели $F(X, C)$ и среды X сообщается блоку адаптации (БА), который вырабатывает сигнал коррекции $\Delta C = \varphi_a(F(X, C), I)$, изменяющий параметры модели в соответствии с (4.1.9) с помощью исполнительного механизма (ИМ).

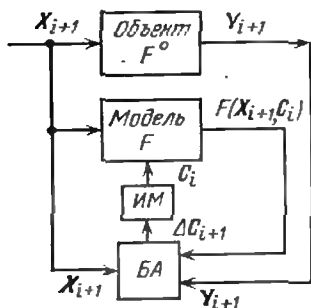


Рис. 4.1.1. Схема адаптивной идентификации

Очевидно, что для реализации адаптивных алгоритмов идентификации вовсе не обязательно использовать реальный масштаб времени. В этом случае роль объекта играет информация I (4.1.4), которая поступает в алгоритм адаптивной идентификации из памяти порциями X_i, Y_i ($i = 1, \dots, N$).

В противоположность адаптивному алгоритму идентификации неадаптивный позволяет получить искомые параметры C сразу, используя всю информацию I (4.1.4), а не путем их постепенного уточнения (ниже будут рассмотрены различные варианты такого алгоритма). Если информация I задана, то задачу идентификации можно решать как адаптивным, так и не адаптивным способом. На первый взгляд может показаться, что неадаптивный алгоритм всегда лучше адаптивного, но

не следует торопиться. Каждый из них имеет свои преимущества и недостатки.

Неадаптивный алгоритм позволяет сразу определить идентифицируемые параметры C , но он сложнее и для его реализации требуются значительные вычислительные мощности — как правило, для этого применяют универсальные ЭВМ.

Адаптивный алгоритм проще: его легко программировать и отлаживать. Кроме того, он может быть эффективно реализован в специализированных вычислительных устройствах. Применяют его обычно для идентификации объектов с изменяющимися свойствами с дрейфующими параметрами в режиме самонастраивающейся модели. Однако его можно применять и для идентификации объекта по всей информации I . В этом случае для эффективной идентификации параметров C , как правило, требуется многократная «прогонка» информации I через адаптивный алгоритм.

4.2. Идентификация статических объектов

Рассмотрим модель статического объекта (4.1.3), где вид функции F' уже определен на стадии структурного синтеза (см. гл. 3). Эта функция может быть задана как аналитически, так и алгоритмически, т. е. в виде инструкций, показывающих, как ее вычислять при всех встречающихся значениях аргументов X и C .

Выход объекта будем считать одномерным ($m=1$), т. е. $Y=y$. Это дает возможность записать модель (4.1.3) в виде

$$y=f(X, C), \quad (4.2.1)$$

где f — заданная на этапе структурного синтеза скалярная функция.

Правомерность такого подхода следует из очевидной декомпозиции задачи (4.1.3), где

$$Y=(y_1, \dots, y_m), \quad F'=(f_1, \dots, f_m),$$

на m задач вида (4.2.1), т. е. декомпонировать модель так, как это показано на рис. 3.3.2. Это означает, что умение идентифицировать параметры модели (4.2.1) с одним выходом позволяет идентифицировать и многовыходовые модели путем m -кратного повторения идентификации модели (4.2.1).

Как видно, эта функция неотрицательна и равна нулю при совпадении правых и левых частей уравнений решаемой системы. При этом чем более правые части уравнения системы (4.2.3) близки к левым, тем меньше значение функции невязки (4.2.4). Это и дает основание считать решением системы (4.2.3) такие значения параметров

$$\mathbf{C}^* = (c^*_1, \dots, c^*_k), \quad (4.2.5)$$

при которых функция невязки минимальна, т. е.

$$Q(\mathbf{C}^*) = \min_{\mathbf{C}} Q(\mathbf{C}).$$

Таким образом, для решения несовместной системы уравнений (4.2.3) достаточно минимизировать функцию суммарной невязки (4.2.4). Эту задачу можно записать в виде

$$Q(\mathbf{C}) \rightarrow \min_{\mathbf{C}}, \quad (4.2.6)$$

которая читается так: «минимизировать функцию $Q(\mathbf{C})$ по $\mathbf{C}$ ».

Пусть решением этой задачи является $\mathbf{C}^*$, для которого при любом $\mathbf{C}$ выполняется очевидное неравенство

$$Q(\mathbf{C}^*) \leq Q(\mathbf{C}), \quad (4.2.7)$$

(напомним, что $\mathbf{C}$ — вектор (4.1.1), т. е. последовательность k компонентов $c_1, \dots, c_k$).

Задача идентификации статических систем сведена таким образом к задаче минимизации некоторой специально построенной функции многих переменных. Как решать такую задачу? Для этого в вычислительной математике разработан ряд эффективных методов, учитывающих специфические особенности минимизируемой функции $Q(\mathbf{C})$.

Нам в дальнейшем придется часто прибегать к решению задачи минимизации (не обязательно функции невязки), поэтому наиболее эффективные современные способы ее решения описаны в приложении. Применяя один из этих способов, можно решить задачу идентификации статического объекта.

Отметим, что для идентификации сложных систем функция суммарной невязки (4.2.4) очень сложна, и не стоит пытаться ее минимизировать «кустарным» способом, т. е. вручную. Такого рода попытки обречены на

провал. Здесь необходимо применение современных ЭВМ и эффективных поисковых методов, описанных в приложении.

Рассмотрим простейший случай, когда функция f имеет вид

$$f(X, C) = \sum_{j=1}^k c_j \varphi_j(X), \quad (4.2.8)$$

где $\varphi_j(X)$ ($j=1, \dots, k$) — заданная система функций многих переменных. Специфика такой структуры заключается в линейности функции относительно искомых параметров $c_1, \dots, c_k$. Эта линейность позволяет свести задачу минимизации (4.2.6) к решению системы линейных алгебраических уравнений, которая, как известно, решается достаточно просто стандартными способами.

Простой вид функции (4.2.8) позволяет решить задачу минимизации (4.2.6), приравняв нулю частные производные функции $Q(C)$, т. е.

$$\frac{\partial Q(C)}{\partial c_l} = 0, \quad l = 1, \dots, k. \quad (4.2.9)$$

Так как f в (4.2.8) является линейной функцией относительно C , то $Q(C)$ в (4.2.4) — квадратичная функция, чем и обуславливается линейность системы уравнений (4.2.9). Действительно, подставляя (4.2.8) в (4.2.4) и дифференцируя, получаем вместо (4.2.9) после очевидных преобразований

$$\sum_{j=1}^k c_j \psi_{jl} - \eta_l = 0, \quad l = 1, \dots, k, \quad (4.2.10)$$

где

$$\begin{aligned} \psi_{jl} &= \sum_{i=1}^N \varphi_j(X_i) \varphi_l(X_i), \\ \eta_l &= \sum_{i=1}^N y_i \varphi_l(X_i), \quad j, l = 1, \dots, k. \end{aligned} \quad (4.2.11)$$

Как видно, (4.2.10) является системой линейных алгебраических уравнений относительно идентифицируемых параметров (4.1.1), которая легко решается стандартными способами (например, методами Гаусса, Крамера и т. д.).

Но здесь не следует обольщаться. Эта простота получена ценой довольно серьезного упрощения: предполагалось, что поведение сложного объекта можно представить в виде (4.2.8), т. е. имеются функции $\psi_i(X)$, с помощью которых можно достаточно точно описать поведение нашего объекта. Такое предположение редко выполняется для сложного объекта. Так, имитационные и ситуационные модели не могут быть идентифицированы этим методом, поскольку подобрать такие функции для них обычно не удастся. Однако этот подход может служить в ряде случаев первым грубым способом получения модели сложного объекта.

Но вернемся к системе уравнений (4.2.10), которая ниже подробно описывается. Рассмотрим матрицу этой системы ($k \times k$):

$$\Phi = \begin{vmatrix} \psi_{11} & \dots & \psi_{1k} \\ \vdots & \ddots & \vdots \\ \psi_{k1} & \dots & \psi_{kk} \end{vmatrix}. \quad (4.2.12)$$

Эту матрицу называют информационной матрицей Фишера. Она симметрична, так как $\psi_{ij} = \psi_{ji}$. Для однозначного решения необходимо, чтобы определитель этой матрицы $|\Phi|$ не был бы равен нулю, т. е.

$$|\Phi| \neq 0. \quad (4.2.13)$$

Этот определитель равен нулю в двух случаях: при $k > N$, т. е. когда недостаточно измерений, и при недостаточной вариабельности входа X , т. е. когда более чем $N - k$ состояний $X_1, \dots, X_N$ линейно зависимы. Реагировать на эту ситуацию можно двояко: либо увеличивать количество наблюдений N , либо (если это не приводит к цели ввиду линейной зависимости) уменьшить размерность контролируемого входа X объекта, т. е. уменьшить число k .

Рассмотрим теперь адаптивную идентификацию статического объекта. Пусть C_i — значения идентифицируемых параметров на i -м шаге адаптивной идентификации. Пусть получена новая информация $I_{i+1} = \langle X_{i+1}, y_{i+1} \rangle$. Эта информация должна изменить C_i на C_{i+1} , т. е. быть источником коррекции параметров:

$$C_{i+1} = C_i + \Delta C_{i+1}.$$

Задача, таким образом, состоит в определении ΔC_{i+1} через I_{i+1} . Для этого образуем локальную невязку выхо-

дов модели и объекта в момент $i+1$:

$$q_{i+1}(\mathbf{C}_i) = f(\mathbf{X}_{i+1}, \mathbf{C}_i) - y_{i+1}. \quad (4.2.14)$$

Очевидно, что величина $\Delta \mathbf{C}_{i+1}$ должна быть такой, чтобы уменьшить квадрат этой невязки. Этого можно легко добиться, если шаг $\Delta \mathbf{C}_{i+1}$ сделать «антиградиентным», т. е.

$$\Delta \mathbf{C}_{i+1} = -\alpha_{i+1} \nabla q^2_{i+1}(\mathbf{C}_i). \quad (4.2.15)$$

Здесь α_{i+1} — некоторый положительный коэффициент (о нем будет сказано позже), а ∇ — «набла», знак оператора градиента функций (см. приложение), которая стоит за этим знаком:

$$\nabla = \left(\frac{\partial}{\partial c_1}, \dots, \frac{\partial}{\partial c_n} \right).$$

(Напомним, что градиент является вектором, направленным в сторону наибольшего увеличения функции. Именно поэтому шаг делается в противоположном антиградиентном направлении функции $q^2(\mathbf{C})$, так как ее нужно минимизировать.)

Определим градиент функции $q^2(\mathbf{C})$:

$$\begin{aligned} \nabla q^2(\mathbf{C}) &= \left(\frac{\partial q^2(\mathbf{C})}{\partial c_1}, \dots, \frac{\partial q^2(\mathbf{C})}{\partial c_k} \right) = \\ &= 2q(\mathbf{C}) \left(\frac{\partial q(\mathbf{C})}{\partial c_1}, \dots, \frac{\partial q(\mathbf{C})}{\partial c_k} \right) = 2q(\mathbf{C}) \nabla q(\mathbf{C}). \end{aligned}$$

Подставляя сюда (4.2.14), получаем

$$\nabla q^2(\mathbf{C}) = 2q(\mathbf{C}) \nabla f(\mathbf{X}, \mathbf{C}).$$

Здесь индексом $\mathbf{C}$ при ∇ обозначен градиент по параметрам $\mathbf{C}$. Это нужно отметить, так как функция зависит еще и от $\mathbf{X}$.

Таким образом, получаем для коррекции параметров на $(i+1)$ -м шаге следующее выражение:

$$\Delta \mathbf{C}_{i+1} = -2\alpha_{i+1} q_{i+1}(\mathbf{C}_i) \nabla_{\mathbf{C}} f(\mathbf{X}_{i+1}, \mathbf{C}_i),$$

где

$$\nabla_{\mathbf{C}} f(\mathbf{X}_{i+1}, \mathbf{C}_i) = \left(\frac{\partial f(\mathbf{X}_{i+1}, \mathbf{C})}{\partial c_1}, \dots, \frac{\partial f(\mathbf{X}_{i+1}, \mathbf{C})}{\partial c_k} \right) \Big|_{\mathbf{C}=\mathbf{C}_i}. \quad (4.2.16)$$

Это выражение легко вычисляется, если известно аналитическое выражение для функции f . Например, подставляя f из (4.2.8), получаем

$$\nabla_C f(X_{i+1}, C_i) = (\varphi_1(X_{i+1}), \dots, \varphi_k(X_{i+1})).$$

Однако для сложных систем функция f определена лишь алгоритмически и вычисление частных производных в (4.2.16) уже не так просто. Для этого приходится воспользоваться конечно-разностными оценками производных (см. приложение)

$$\frac{\partial f(X, C)}{\partial c_l} = \frac{1}{2g} [f(X, C + g e_l) - f(X, C - g e_l)], \quad (4.2.17)$$

$$\frac{\partial f(X, C)}{\partial c_l} = \frac{1}{g} [f(X, C + g e_l) - f(X, C)], \quad (4.2.18)$$

где $g > 0$ — база оценки, а e_l — единичный вектор, направленный вдоль l -й оси. Легко заметить, что при $g \rightarrow 0$ эти оценки стремятся к точному значению частной производной.

Теперь о выборе параметра α . При малом его значении коррекция будет невелика. Но его нельзя делать и слишком большим, так как при этом невязка увеличится. Поведение величины невязки q^2 в зависимости от α показано на рис. 4.2.1. Хорошо видно, что при $\alpha = \alpha^*$

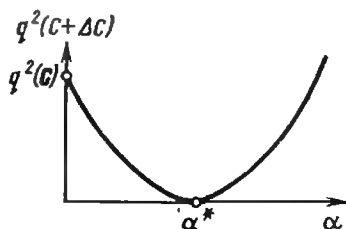


Рис. 4.2.1. К определению параметра α

невязка минимальна и равна нулю. Этим обстоятельством можно воспользоваться для определения оптимального значения α . Для этого достаточно решить задачу однопараметрической минимизации

$$q^2(C - 2\alpha q(C) \nabla_C f(X, C)) \rightarrow \min_{\alpha > 0}, \quad (4.2.19)$$

т. е. минимизировать квадрат невязки по параметру α . Но, зная, что минимальное значение невязки при отсутствии помех равно нулю, эту задачу можно решить проще, приравняв невязку нулю:

$$q(C - 2\alpha q(C) \nabla_C f(X, C)) = 0. \quad (4.2.20)$$

Решение этого уравнения и дает оптимальное значение α^* . Для линейного случая (4.2.8) получаем из (4.2/20) после очевидных преобразований

$$\alpha^* = 1/2 \left[1 + \sum_{j=1}^n (x_j^{l+1})^2 \right]^{-1},$$

где x_j — j -я координата вектора входа X . В более сложных задачах, когда модель f определяется алгоритмически (например, для имитационных моделей), оптимальное значение параметра α^* следует определять из решения задачи однопараметрической минимизации (4.2.19), различные способы решения которой приведены в приложении.

Теперь перейдем к идентификации динамических моделей.

4.3. Идентификация динамических моделей

Как известно, динамический объект отличается от статического тем, что его состояние Y_i определяется не только состоянием его входов X_i , но и собственным состоянием в предыдущий момент времени Y_{i-1} . Это и есть та самая «память» объекта, которая делает его инерционным (в узком механическом смысле) и динамическим (в широком). Таким образом, дискретную модель динамического объекта на стадии идентификации можно записать в общем виде:

$$Y_i = F(Y_{i-1}, X_i, C), \quad (4.3.1)$$

где F —определенная на стадии структурного синтеза функция, которая может быть задана как аналитически, так и алгоритмически (последняя форма задания более распространена для сложных объектов управления). Задача идентификации состоит в том, чтобы определить параметры $C = (c_1, \dots, c_n)$ модели (4.3.1) по наблюдениям I входа и выхода объекта (см. (4.1.4)).

Заметим, что разница в выражениях (4.3.1) и (4.1.2) состоит в том, что в (4.1.2) F —оператор, а в (4.3.1) F —функция (отсутствие в (4.3.1) управления U несущественно, так как оно является таким же входом, как и X , и поэтому на стадии идентификации может быть объединено с X , если, разумеется, изменяется в режиме нормальной эксплуатации объекта).

Как и в предыдущем параграфе, будем рассматривать модель с одним выходом ($m=1$), что, как известно, не снижает общности рассуждений. Пусть для простоты объект имеет один вход ($n=1$) (ниже будет показано, как можно снять это ограничение). Исходная информация (4.2.2) о функционировании объекта принимает вид

$$I = \langle x_i, y_i \rangle, \quad i = 1, \dots, N, \quad (4.3.2)$$

где x_i и y_i — значения входа и выхода объекта в i -й момент времени t_i , т. е. $x_i = x(t_i)$, $y_i = y(t_i)$.

Если состояние Y динамического объекта определяется его выходом в данный и p предыдущих моментов времени $t_i, \dots, t_{i-p}$, то такой объект называют динамическим объектом p -го порядка. Очевидно, что состояние этого объекта определяется значениями его выхода, т. е. вектором

$$Y'_i = (y_i, y_{i-1}, \dots, y_{i-p}). \quad (4.3.3)$$

Теперь модель (4.3.1) динамического объекта p -го порядка представляется в виде

$$y_i = f(Y'_i, X'_i, C), \quad (4.3.4)$$

где f — заданная на этапе структурного синтеза скалярная функция, вектор Y'_i определен в (4.3.3), а X'_i — вектор возмущения объекта:

$$X'_i = (x_i, x_{i-1}, \dots, x_{i-l}), \quad (4.3.5)$$

который связывает поведение объекта с предыдущими l состояниями среды. Числа p и l характеризуют структуру объекта и должны быть определены на стадии структурного синтеза модели.

Рассмотрим сначала неадаптивный метод идентификации. В этом случае очевидно, что параметры C должны быть такими, чтобы суммарная невязка выходов модели и объекта

$$Q(C) = \sum_{i=s}^N [f(Y'_{i-1}, X'_i, C) - y_i]^2 \quad (4.3.6)$$

была бы минимальной, где y_i , Y'_{i-1} , X'_i определяются наблюдениями (4.3.2), а

$$s = \max(p+2, l+1), \quad (4.3.7)$$

т. е. является максимальным из двух чисел $p+2$ и $l+1$.

Суммирование невязки (4.3.6), начиная с s , а не с 1, связано с тем, что необходимо иметь l предыдущих значений x_i и y_i .

Теперь задача идентификации параметров C сводится к минимизации функции (4.3.6):

$$Q(C) \rightarrow \min_C.$$

Решить эту задачу можно одним из поисковых методов оптимизации, изложенных в приложении. Полученные значения параметров, обеспечивающие минимум функции суммарной невязки (4.3.6), и являются решением задачи идентификации динамического объекта, описываемого моделью (4.3.4).

Например, для линейного динамического объекта, модель которого имеет вид (3.4.5); суммарная функция невязки (4.3.6) имеет вид

$$Q(C) = \sum_{l=s}^N \left[\sum_{r=1}^p c_r y_{i-r} + \sum_{j=1}^{l+1} c_{p+j} x_{i-j+1} - y_i \right]^2,$$

где число идентифицируемых параметров $k = p + l + 1$. Как видно, это квадратичная форма, минимизация которой сводится к решению системы линейных алгебраических уравнений:

$$\frac{\partial Q(C)}{\partial c_z} = 0, \quad z = 1, \dots, k = p + l + 1.$$

Обобщим теперь этот подход на объекты со многими входами ($n \geq 1$): $X = (x_1, \dots, x_n)$. В этом случае под X'_i в выражении (4.3.4) следует понимать значения входа X при всех l предыдущих моментах времени:

$$X'_i = (X_i, X_{i-1}, \dots, X_{i-l}),$$

где X_r ($r = i, \dots, i-l$) — значение входа X объекта в r -й момент времени t_r . Исходная информация в этом случае имеет вид (4.2.2).

Таким образом, многомерность входа объекта в принципе не усложняет алгоритма решения задачи идентификации, а делает его лишь более громоздким. Так, при линейной зависимости (4.3.4) по C число идентифицируемых параметров увеличивается с $k = p + l + 1$ (при одномерном входе) до $k = p + n(l + 1)$ (при n -мерном входе X).

Теперь рассмотрим адаптивный метод идентификации. Он состоит в минимизации на каждом шаге квадрата локальной невязки выходов модели и объекта:

$$q^2(C_{i-1}) = [f(Y'_{i-1}, X'_{i-1}, C_{i-1}) - y_i]^2. \quad (4.3.8)$$

Это можно осуществить соответствующим выбором коррекции ΔC_i параметров C так же, как это было сделано в предыдущем параграфе, но применительно к (4.3.8):

$$\Delta C_i = -\alpha_i \nabla C q^2(C_{i-1}),$$

где ∇ — по-прежнему оператор градиента, а параметр α_i выбирается оптимальным из соображений минимизации локальной невязки (4.3.8):

$$q^2(C_{i-1} - \alpha_i \nabla C q^2(C_{i-1})) \rightarrow \min_{\alpha_i > 0} \Rightarrow \alpha_i^*.$$

В линейном случае (3.4.5) и при отсутствии помех получаем

$$\nabla C q^2(C_{i-1}) = 2q(C_{i-1}) (y_{i-1}, \dots, y_{i-p-1}, x_i, \dots, x_{i-l});$$

$$\alpha_i^* = 1/2 \left(\sum_{j=1}^l x_{i-j}^2 + \sum_{s=1}^{p+1} y_{i-s}^2 \right)^{-1}$$

— оптимальное значение параметра α , при котором невязка $q(C_{i-1} + \Delta C_i)$ становится равной нулю.

Заметим в заключение этой главы, что процесс идентификации параметров $c_1, \dots, c_k$ модели объекта управления всегда связан с процедурой минимизации «несоответствия» (невязки) модели и объекта, которое вычисляется как разность реакции (выходов) модели и объекта на одинаковый вход. Эта минимизация может производиться различными методами поиска (некоторые из них описаны в приложении) и заканчивается определением параметров модели $C^* = (c_1^*, \dots, c_k^*)$, которые наиболее соответствуют имеющейся информации (4.1.4) о работе объекта. Однако это совсем не означает, что модель адекватна объекту. Это лишь наилучшее приближение при выбранной структуре модели. Вопрос об адекватности должен решаться на стадии выбора структуры модели и по величине минимальной невязки (4.3.6). Чем больше априорной информации используется при определении структуры модели, тем больше шансов получить адекватную модель.

Идентификация как процесс определения параметров модели по наблюдениям, полученным в режиме нормального функционирования объекта, исследован довольно хорошо. Для первого ознакомления с задачей идентификации читателю рекомендуются книги [1 и 2]. Идентификация промышленных объектов является наиболее важной практической задачей современной техники управления, и поэтому ей уделяется большое внимание. Методы идентификации такого рода объектов описаны в работах [3—5]. Достаточно полное (хотя и не простое) изложение основных методов идентификации и их некоторых приложений содержится в монографии [6].

Адаптивная идентификация, рассмотренная в § 4.2, является в большинстве случаев наиболее эффективным подходом к решению практических задач идентификации. Ей посвящены работы [7, 8], а в книге [9] описано очень интересное приложение адаптивной идентификации для решения задач управления точностью прокатки труб, реализованное на производстве, за которое ее авторы в 1976 г. получили Государственную премию.

СПИСОК ЛИТЕРАТУРЫ

1. Райбман Н. С. Что такое идентификация? — М.: Наука, 1970.
2. Растринин Л. А., Маджаров Н. Е. Введение в идентификацию объектов управления. — М.: Энергия, 1977.
3. Балакирев С. В., Дудников Е. Г., Цирлин А. М. Экспериментальное определение динамических характеристик промышленных объектов. — М.: Энергия, 1967.
4. Ордынцев В. М. Математическое описание объектов автоматизации. — М.: Машиностроение, 1966.
5. Райбман Н. С., Чадаев В. М. Построение моделей производства. — М.: Энергия, 1975.
6. Эйкхофф П. Основы идентификации систем управления: Пер. с англ. — М.: Мир, 1975.
7. Райбман Н. С., Чадаев В. М. Адаптивные модели в системах управления. — М.: Сов. радио, 1966.
8. Райбман Н. С., Чадаев В. М. Адаптивные системы управления технологическими процессами: методика/ Ии-т проблем управления. — М., 1972.
9. Адаптивное управление точностью прокатки труб/ Под ред. Ф. А. Данилова и Н. С. Райбмана. — М.: Металлургия, 1973.

ГЛАВА 5

ПЛАНИРОВАНИЕ ЭКСПЕРИМЕНТОВ

Этап планирования экспериментов с объектом прежде всего связан с необходимостью определить параметры влияния на выход Y тех входов объекта, которые не варьируются при его нормальной работе, т. е. без управления. Очевидно, что такими входами прежде всего

являются определенные на этапе структурного синтеза управляемые входы U . На этапе структурного синтеза получена зависимость

$$Y=F(X, U, C), \quad (5.0.1)$$

где F — определенный оператор модели, а $C=(c_1, \dots, c_k)$ — неизвестные параметры объекта, которые должны быть определены на этапе планирования экспериментов.

Другая задача планирования экспериментов возникает при определении структуры модели объекта управления. В этом случае необходимо построить такой эксперимент, чтобы полученная информация дала бы возможность выбрать наилучшую модель из заданного набора конкурирующих моделей и при этом определить ее параметры. Необходимость идти на такую нежелательную операцию, как эксперименты с объектом (напомним, что мы строим систему управления сложным объектом, который «не любит» экспериментов), связана с тем, что иного пути здесь нет.

Математически неизбежность экспериментов с объектом вытекает из следующего простого рассуждения. Для определения параметров $C=(c_1, \dots, c_k)$ модели (5.0.1) необходимо иметь исходную информацию о состоянии входов и выходов объекта в виде троек:

$$I=\langle X_i, U_i, Y_i \rangle, \quad i=1, \dots, N. \quad (5.0.2)$$

Но на стадии синтеза системы управления вход U не изменяется, т. е. $U_i=\text{const}$, так как объект еще не управляется. Это и исключает возможность определения параметров, характеризующих связь входа U с выходом Y на стадии идентификации. Для того чтобы определить эти параметры, необходимо варьировать вход U и фиксировать при этом поведение объекта, т. е. экспериментировать с ним.

Будем для простоты записи рассматривать случай

$$Y=F(U, C). \quad (5.0.3)$$

Общий случай (5.0.1) легко получится из (5.0.3) при дополнительном фиксировании (записи) состояния X среды в процессе экспериментов.

Задача планирования экспериментов заключается в таком минимальном изменении входа U , при котором

полученная информация Y давала бы возможность наилучшим образом оценить значения параметров C . Рассмотрим эту задачу подробнее.

5.1. Основные понятия теории планирования экспериментов

Теория планирования экспериментов возникла как раздел математической статистики, и поэтому в ней сложилась терминология, отличная от терминологии теории управления (это следует иметь в виду при чтении литературы по планированию).

Под объектом планирования будем подразумевать многополюсник с q управляемыми входами $U = (u_1, \dots, u_q)$. Вид (структура) оператора F (5.0.3) уже определен на стадии структурного синтеза модели, рассмотренного в гл. 3. Он имеет стохастическую структуру вида (3.4.9).

Для решения задачи планирования экспериментов следует организовать такое изменение входов U объекта, чтобы можно было эффективно определить параметры C модели и, если нужно, скорректировать структуру St модели F .

Схема процесса планирования экспериментов показана на рис. 5.1.1. Она напоминает схему системы управления, рассмотренную во введении (см. рис. В.2.1). Это не удивительно, так как планирование эксперимента действительно является управлением объекта, но цели этого управления специфичны — определение параметров модели объекта, а точнее, построение модели объекта управления (возможна и коррекция структуры модели). Как видно из рис. 5.1.1, роль управляющего устройства выполняет процедура планирования эксперимента, в которой указывается, каким образом следует изменять вход U в процессе эксперимента. Сам эксперимент однозначно определяется планом $\tilde{U}$, представляющим собой перечень состояний входа U , которые должны быть реализованы в объекте. Так, для статического объекта план $\tilde{U}$ имеет вид

$$\tilde{U} = \langle U_1, \dots, U_N \rangle, \quad (5.1.1)$$

где U_i — i -й эксперимент плана, т. е. i -е состояние входа:

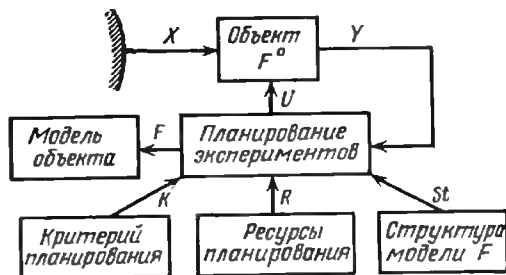
$$U_i = (u_{1i}, \dots, u_{qi}), \quad i = 1, \dots, N.$$

План (5.1.1) полностью определяется так называемой *матрицей планирования*

$$\tilde{U} = \begin{bmatrix} u_1 \\ \vdots \\ u_N \end{bmatrix} = \begin{bmatrix} u_{11} & \dots & u_{q1} \\ \vdots & & \vdots \\ u_{1N} & \dots & u_{qN} \end{bmatrix}. \quad (5.1.2)$$

Для динамического объекта планом является векторная функция времени $\tilde{U} = U(t)$, т. е. зависимость входа U от времени.

Рис. 5.1.1. Схема процесса планирования экспериментов с объектом



Цели планирования эксперимента, на основе которых строится план эксперимента $\tilde{U}$, обычно определяются двумя факторами: ресурсом и критерием планирования.

Ресурсы планирования R состоят из выделяемых на эксперимент средств P (временных, материальных, штатных и т. д.), а также области (зоны) планирования Ω , определяющей пределы изменения входа U в процессе планирования:

$$\tilde{U} \in \Omega. \quad (5.1.3)$$

Таким образом,

$$R = \langle P, \Omega \rangle. \quad (5.1.4)$$

Например, $P = N$ — объем плана, т. е. число экспериментов, которое можно произвести с объектом. При этом Ω : $a_i \leq u_i \leq b_i$ ($i=1, \dots, q$) — гиперпараллелепипед в пространстве состояний входа U , где значения границ a_i и b_i ($i=1, \dots, q$) должны быть заданы.

Критерий планирования определяет эффективность плана $\tilde{U}$. Этот критерий естественно зависит от плана $\tilde{U}$ и структуры модели F объекта:

$$K = K(\tilde{U}, F). \quad (5.1.5)$$

В качестве такого критерия планирования обычно выбирают характеристики процесса планирования, определяющие точность оценки параметров S модели. Например, это может быть сумма или произведение дисперсий оценок определяемых параметров или максимальная дисперсия и т. д. Минимизация такого рода критерия с помощью плана $\tilde{U}$ является, очевидно, целесообразной мерой при планировании экспериментов.

Иногда может возникнуть ситуация, когда необходимо минимизировать не один, а несколько (p) критериев. Тогда следует решить многокритериальную задачу

$$K_i(\tilde{U}, F) \rightarrow \min, \quad i = 1, \dots, p, \quad (5.1.6)$$

$$\tilde{U} \in R$$

Для ее решения необходимо привлекать дополнительную информацию об объекте (способы решения такого рода задач описаны в приложении). Основой для решения здесь являются различного рода компромиссы между критериями. Поэтому планы, синтезированные при решении многокритериальной задачи (5.1.6), называют компромиссными.

Как видно, цели планирования определяются исходя из специфики объекта управления и целей его эксплуатации и управления. Нельзя забывать, что сложный объект, которым мы занимаемся, существует и функционирует независимо от целей будущего управления, и поэтому вмешательство в его работу до управления им должно происходить с большой «оглядкой». Прежде всего это касается выбора области планирования Ω .

При выборе размеров этой области неизбежен конфликт между «хозяйном» объекта, т. е. лицом, ответственным за его нормальное функционирование, и проектировщиком системы управления. Если первый стремится минимизировать размер области Ω , чтобы минимально нарушать нормальную работу объекта, то второй заинтересован в максимизации размеров Ω , так как при этом эффективность эксперимента $\tilde{U}$ бывает наибольшей. Действительно, чем больше отличаются друг от друга состояния входа (5.1.1) при эксперименте, тем больше будет получено информации об объекте. Указанный конфликт обычно решается с помощью разумного компромисса между обеими сторонами и приводит

к выбору конкретных очертаний области планирования Ω .

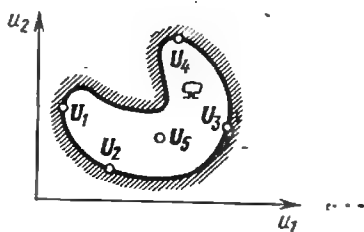
Таким образом, реализация целей планирования связана с решением следующей экстремальной задачи:

$$K(\tilde{U}, F) \rightarrow \min_{\tilde{U} \in R} \Rightarrow \tilde{U}^*_K, \quad (5.1.7)$$

т. е. варьируя план $\tilde{U}$ в пределах выделенных ресурсов R , следует минимизировать заданный критерий при определенной ранее структуре модели F . Полученный при этом план $\tilde{U}^*_K$ естественно назвать K -оптимальным

Рис. 5.1.2. Зона планирования Ω в пространстве управляемых входов $\{U\}$.

Точки u_i обозначены эксперименты, образующие план $\tilde{U}^*$.



планом с ресурсами R для объекта с моделью F . При изменении указанных факторов (критерия, ресурса и объекта) оптимальный план $\tilde{U}^*_K$ также изменяется.

На рис. 5.1.2 показан пример оптимального плана $\tilde{U}^*$, состоящего из $N=5$ экспериментов для статического объекта (F — функция) с заданной структурой. В этом случае каждый эксперимент характеризуется точкой в пространстве входов $\{U\}$, а план $\tilde{U}^*_K$ представляет собой набор из пяти точек, принадлежащих заданной области (зоне) планирования Ω :

$$\tilde{U}^*_K = \langle U_1, \dots, U_5 \rangle, \quad U_i \in \Omega, \quad i = 1, \dots, 5,$$

причем их расположение минимизирует заданный критерий K .

Заметим, что очень часто управление U имеет дискретный характер. Например, i -я рукоятка управления объектом может находиться только в двух положениях: «включено» ($u_i=1$), «выключено» ($u_i=0$); промежуточные положения при этом исключаются. Учет этой особенности объекта производится соответствующим опре-

делением множества Ω , которое в дискретном случае состоит из конечного числа элементов. Так, если i -й параметр управления u_i может принимать лишь k_i значений, то число элементов множества Ω равно

$$k_1 k_2 \dots k_q = \prod_{i=1}^q k_i.$$

Для двоичного (бинарного) управления ($k_i=2, i=1, \dots, q$) получаем 2^q элементов этого множества.

Основная проблема планирования экспериментов, таким образом, заключается в эффективном решении задачи (5.1.6), т. е. в создании оптимальных планов $\tilde{U}^*$ на допустимом ресурсном множестве R .

Будем рассматривать только статические объекты, для моделей которых F является функцией. В этом случае план $\tilde{U}$ представляет собой набор из N точек $U_1, \dots, U_N$ в пространстве варьируемых параметров $\{U\}$, причем все точки принадлежат зоне планирования: $U_i \in \Omega$ ($i=1, \dots, N$).

Планирование экспериментов для динамического объекта (когда F — оператор) представляет собой функцию $U=U(t) \in \Omega$, описывающую траекторию в зоне планирования Ω , причем эта траектория минимизирует выбранный критерий K , который в данном случае является функционалом *). Подходы к решению таких задач рассмотрены в следующей главе.

Теперь рассмотрим способы формулировки критерия планирования K .

5.2. Критерии планирования

Рассмотрим эксперимент, целью которого является наиболее достоверное определение параметров $C=(c_1, \dots, c_k)$. Однако понятие «достоверности» слишком широко и размыто. Поэтому в каждом конкретном случае формулировка критерия определенным образом связана с целями управления Z^* , но всегда зависит от точ-

*) Напомним, что функционалом является преобразование функции в число, например:

$$\int_a^b |U(t)|^2 dt, \quad \max_{0 \leq t \leq T} |U(t)| \text{ и т. д.}$$

ностных характеристик искомых параметров. Такими характеристиками являются, например, дисперсия параметров C

$$D_{ii} = M[(c_i - \bar{c}_i)^2], \quad i = 1, \dots, k,$$

(где $\bar{c}_i$ — среднее значение параметра c_i , а M — знак математического ожидания) и коэффициенты корреляции этих параметров

$$D_{ij} = M[(c_i - \bar{c}_i)(c_j - \bar{c}_j)], \\ i, j = 1, \dots, k, \quad i \neq j.$$

Эти характеристики объединяются так называемой дисперсионной матрицей параметров C , которую обычно обозначают так:

$$D_C = \begin{bmatrix} D_{11} & \dots & D_{1k} \\ \vdots & \ddots & \vdots \\ D_{k1} & \dots & D_{kk} \end{bmatrix}. \quad (5.2.1)$$

Как видно, дисперсионная матрица (5.2.1) полностью определяет статистические свойства искомых параметров C .

Естественно задать вопрос, как определять дисперсионную матрицу? В общем случае это довольно сложная задача. Однако для моделей, линейных относительно искомых параметров C , т. е. моделей вида (3.4.3), известно, что дисперсионная матрица обратна информационной матрице Фишера (4.2.12), т. е.

$$D_C = \sigma^2 \Phi^{-1}. \quad (5.2.2)$$

Это означает, что каждый элемент дисперсионной матрицы равен

$$D_{ij} = \sigma^2 \Phi_{ji} / |\Phi|, \quad i, j = 1, \dots, k. \quad (5.2.3)$$

Здесь Φ_{ji} — алгебраическое дополнение элемента ψ_{ji} матрицы Φ , которое получается из Φ вычеркиванием j -й строки и i -го столбца и умножением $(-1)^{i+j}$; $|\Phi|$ — определитель матрицы Φ ; σ^2 — дисперсия случайной помехи, накладывающейся на выход объекта в соответствии со структурой (3.4.9).

Таким образом, зная матрицу Φ , легко по (5.2.3) получить дисперсионную матрицу D_C , с помощью которой можно сформулировать любые необходимые критерии планирования.

Наиболее распространенным является так называемый D -критерий, равный определителю дисперсионной матрицы (5.2.1), т. е. обобщенной дисперсии:

$$K = |D_c|. \quad (5.2.4)$$

План, минимизирующий этот критерий, называют D -оптимальным.

Вообще говоря, критерием планирования может быть любая функция элементов дисперсионной матрицы $K = \varphi(D_c)$. Например, K может быть максимальной дисперсией ($K = \max_{i=1, \dots, q} D_{ii}$). План, минимизирующий эту дисперсию, называют E -оптимальным. Если K — сумма дисперсий искомых параметров ($K = \sum_{i=1}^q D_{ii}$), т. е. след дисперсионной матрицы*), то соответствующий план называют A -оптимальным, и т. д.

Особое место среди многочисленных планов экспериментов занимают ортогональные планы, для которых информационная матрица Фишера Φ диагональна. С помощью ортогонального плана очень упрощается процедура вычисления параметров C :

$$c_i = \psi_{ii} / \eta_i, \quad i = 1, \dots, k,$$

где ψ_{ii} и η_i определены в (4.2.11). Однако не в этом основное преимущество ортогонального плана, а во взаимной некоррелированности оценок искомых параметров. Действительно, диагональность информационной матрицы Φ приводит к диагональности дисперсионной матрицы D , откуда следует равенство нулю всех коэффициентов корреляции:

$$D_{ij} = 0, \quad i \neq j = 1, \dots, k,$$

что означает указанную взаимную некоррелированность получаемых оценок искомых параметров C . Это обстоятельство часто бывает очень важным при синтезе модели.

Теперь рассмотрим наиболее распространенное D -оптимальное планирование экспериментов.

*) Напомним, что следом матрицы называют сумму ее диагональных членов.

5.3. D-оптимальное планирование

Как сказано выше, синтез D -оптимального плана сводится к решению задачи минимизации определителя дисперсионной матрицы (5.2.1) или в силу (5.2.2) к максимизации определителя информационной матрицы Φ :

$$|\Phi(\tilde{U})| \rightarrow \max_{\tilde{U} \in \Omega} \Rightarrow \tilde{U}_D^*. \quad (5.3.1)$$

Полученный при этом план $\tilde{U}_D^*$ и называют D -оптимальным.

Построим D -оптимальный план для линейного одномерного объекта:

$$y = c_1 + c_2 u. \quad (5.3.2)$$

Элементы информационной матрицы в этом случае можно было бы определить по формуле (4.2.11), где $\varphi_1 = 1$, $\varphi_2 = u$. Однако из соображений наглядности произведем все необходимые выкладки для рассматриваемого простого случая ($n = q = 1$). Для того чтобы воспользоваться методом наименьших квадратов при определении параметров c_1 и c_2 модели (5.3.2), введем условно исходную информацию в виде каких-то результатов эксперимента с объектом:

$$I = \langle u_i, y_i \rangle \quad (i = 1, \dots, N), \quad (5.3.3)$$

где $\tilde{U} = \langle u_1, \dots, u_N \rangle$ — искомый план эксперимента.

Суммарная невязка выходов модели (5.3.2) и объекта (5.3.3)

$$Q(c_1, c_2) = \sum_{i=1}^N [c_1 + c_2 u_i - y_i]^2.$$

Минимизируя это выражение в соответствии с методом наименьших квадратов по c_1 и c_2 , получаем систему двух уравнений

$$\begin{aligned} \frac{\partial Q}{\partial c_1} &= 2 \sum_{i=1}^N (c_1 + c_2 u_i - y_i) = 0, \\ \frac{\partial Q}{\partial c_2} &= 2 \sum_{i=1}^N (c_1 + c_2 u_i - y_i) u_i = 0. \end{aligned}$$

Эту систему удобно преобразовать к виду

$$\begin{aligned} c_1 + c_2 \bar{u} &= \frac{1}{N} \eta_1, \\ c_1 \bar{u} + c_2 \bar{u}^2 &= \frac{1}{N} \eta_2, \end{aligned}$$

где

$$\bar{u} = \frac{1}{N} \sum_{i=1}^N u_i; \quad \bar{u}^2 = \frac{1}{N} \sum_{i=1}^N u_i^2.$$

Откуда видно, что определитель информационной матрицы имеет вид

$$|\Phi(\tilde{u})| = \begin{vmatrix} 1 & \bar{u} \\ \bar{u} & \bar{u}^2 \end{vmatrix} = \bar{u}^2 - (\bar{u})^2. \quad (5.3.4)$$

Теперь решим задачу синтеза D -оптимального плана, максимизируя это выражение по элементам плана $\tilde{U}$ параметром u_i ($i=1, \dots, N$). Для этого необходимо решить минимизационную задачу

$$\bar{u}^2 - (\bar{u})^2 \rightarrow \max, \\ u_i \in \Omega \quad (i=1, \dots, N),$$

где $\Omega: [a, b]$, т. е. задача состоит в выборе таких N чисел u_i внутри отрезка $[a, b]$, т. е. $a \leq u_i \leq b$, которые минимизировали бы квадратичную форму (5.3.4).

Эта задача решается сразу путем следующего простого рассуждения. Легко заметить, что выражение (5.3.4) представляет собой не что иное, как дисперсию точек плана $\tilde{U} = (u_1, \dots, u_N)$. Максимизация дисперсии N точек на отрезке требует, чтобы они были расположены на концах этого отрезка, причем половина на одном, а половина на другом конце.

Следовательно, D -оптимальный план $\tilde{U}^*_D = (u^*_1, \dots, u^*_N)$ в этом случае при четном N выглядит так:

$$u^*_i = \begin{cases} a & \text{для } i \leq N/2, \\ b & \text{для } i > N/2; \end{cases}$$

при нечетном N возможны два эквивалентных плана:

$$u^*_i = \begin{cases} a & \text{для } i \leq (N-1)/2, \\ b & \text{для } i > (N-1)/2 \end{cases}$$

или

$$u^*_i = \begin{cases} a & \text{для } i \leq (N+1)/2, \\ b & \text{для } i > (N+1)/2. \end{cases}$$

Заметим, что в данном случае информационная матрица (5.3.4) недиагональна, план неортогонален и, следовательно, получаемые при этом оценки параметров c_1 и c_2 коррелированы. Действительно, их дисперсионная матрица равна

$$DC = \sigma^2 \Phi^{-1} = \sigma^2 \begin{vmatrix} \bar{u}^2 & -\bar{u} \\ -\bar{u} & 1 \end{vmatrix}. \quad (5.3.5)$$

Однако это неудобство легко преодолеть. План $\tilde{U}^*$ можно ортогонализировать, изменив систему координат так, чтобы получить $\bar{u}=0$,

что, как видно из (5.3.5), делает матрицу Φ диагональной. Для этого достаточно перейти к центрированной системе координат, т. е. расположить начало координат в центре зоны планирования Ω :

$$u' = -\frac{a+b}{b-a} + \frac{2}{b-a}u,$$

для которых $\bar{u}'=0$, $(\bar{u}')^2=1$ и матрица получает требуемый вид

$$\Phi' = \left\| \begin{array}{cc} 1 & 0 \\ 0 & 1 \end{array} \right\|.$$

План в этом случае записывается так:

$$\tilde{U}^*_D = \underbrace{(-1, \dots, -1, 1, \dots, 1)}_N,$$

где
$$k = \begin{cases} N/2 & \text{для четных } N, \\ (N+1)/2 & \text{для нечетных } N. \end{cases}$$

Как видно, от выбора системы координат существенно зависят корреляционные свойства определяемых параметров.

Применяя аналогичные рассуждения для D -оптимального плана эксперимента в случае линейной многомерной модели $y=c_1+c_1u_2+\dots+c_{q+1}u_q$ в области Ω : $-1 \leq u_i \leq 1$ (гиперкуб в пространстве $\{U\}$), получим D -оптимальный план $\tilde{U}^*_D$ с координатами $u^*_j=\pm 1$ ($j=1, \dots, q$), т. е. точки U_i ($i=1, \dots, N$) должны располагаться в углах гиперкуба, причем так, чтобы план $\tilde{U}$ при этом был ортогональным. Условия ортогональности

$$\sum_{i=1}^N u_{ji}u_{li} = 0 \quad \text{при } i \neq l$$

состоят в том, что произведения двух любых столбцов матрицы планирования (5.1.2) равны нулю. Такая матрица называется ортогональной относительно столбцов. На рис. 5.3.1 приведены примеры различных планов для $n=2$, когда Ω — квадрат. Их матрицы планирования соответственно равны

$$\tilde{U}^*_D = \tilde{U}^0 = \left\| \begin{array}{cc} 1 & 1 \\ -1 & 1 \\ -1 & -1 \\ 1 & -1 \end{array} \right\|; \quad \tilde{U}^0 = \left\| \begin{array}{cc} 0 & 1 \\ -1 & 0 \\ 0 & -1 \\ 1 & 0 \end{array} \right\|;$$

$$\tilde{U}_D^* = \begin{vmatrix} 1 & 1 \\ 1 & 1 \\ -1 & 1 \\ -1 & -1 \\ 1 & -1 \end{vmatrix}; \quad \tilde{U}_D^* = \begin{vmatrix} 1 & 1 \\ -1 & -1 \\ 1 & -1 \end{vmatrix}.$$

Теперь рассмотрим пример квадратичной зависимости F в простейшем одномерном случае

$$y = c_1 + c_2 u + c_3 u^2,$$

где функции разложения (4.2.8): $\varphi_1(u) = 1$; $\varphi_2(u) = u$; $\varphi_3(u) = u^2$.

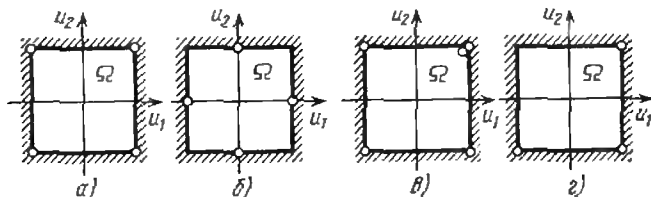


Рис. 5.3.1. Примеры планов:

a — D -оптимальный ортогональный ($N=4$), $б$ — ортогональный не D -оптимальный ($N=4$), $в$ — D -оптимальный неортогональный ($N=5$), $г$ — то же, но для $N=3$

Информационная матрица Фишера (4.2.12) в этом случае, как легко показать, приобретает вид

$$\Phi = \begin{bmatrix} 1 & \bar{u} & \bar{u}^2 \\ \bar{u} & \bar{u}^2 & \bar{u}^3 \\ \bar{u}^2 & \bar{u}^3 & \bar{u}^4 \end{bmatrix}, \quad (5.3.6)$$

где

$$\bar{u}^j = \frac{1}{N} \sum_{i=1}^N u_i^j, \quad j = 1, 2, 3, 4.$$

Любопытно, что в этом случае (при заданных переменных) ортогонального плана не существует, так как при $\bar{u} = \bar{u}^2 = \bar{u}^3 = 0$ определитель $\{\Phi\}$ становится равным нулю, что исключает возможность определения C . Поэтому для простоты синтеза плана естественно потребовать:

$$\bar{u} = \bar{u}^3 = 0, \quad (5.3.7)$$

т. е. план должен быть симметричным относительно центра зоны планирования Ω .

Пусть $\Omega: -1 \leq u \leq 1$, тогда определитель информационной матрицы Фишера

$$|\Phi| = \begin{vmatrix} 1 & 0 & \bar{u}^2 \\ 0 & \bar{u}^2 & 0 \\ \bar{u}^2 & 0 & \bar{u}^4 \end{vmatrix} = \bar{u}^2 [\bar{u}^4 - (\bar{u}^2)^2].$$

Теперь построим оптимальные планы с различным объемом N , т. е. будем варьировать число точек плана.

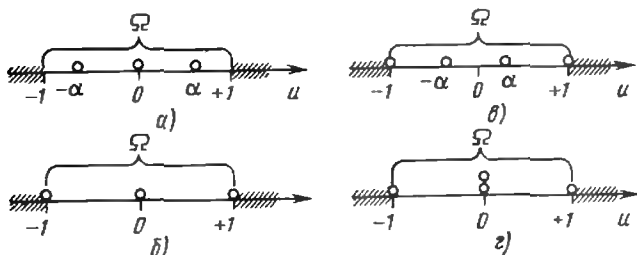


Рис. 5.3.2. К построению D -оптимальных планов на отрезке для квадратичной зависимости F :

$a, в$ — неоптимальные; $б, г$ — оптимальные планы

Пусть $N=3$. Симметричный план в этом случае определяется одним параметром α и имеет вид

$$\tilde{U} = \langle -\alpha, 0, \alpha \rangle$$

(рис. 5.3.2, а). Для второго и четвертого моментов получаем

$$\bar{u}^2 = 2/3 \alpha^2; \quad \bar{u}^4 = 2/3 \alpha^4$$

и задача синтеза оптимального плана сводится к максимизации выражения

$$|\Phi| = 4/27 \alpha^6 \rightarrow \max \Rightarrow \alpha^* = 1, \quad -1 \leq \alpha \leq 1$$

решение которой очевидно. Таким образом, D -оптимальный план в этом случае имеет вид (см. рис. 5.3.2, б) $\tilde{U}^*_{D} = \langle -1, 0, +1 \rangle$.

Пусть $N=4$. Тогда, фиксируя две очевидные точки плана на его краях ($u = \pm 1$), рассмотрим симметричный план с одним неизвестным параметром $\alpha > 0$: $\tilde{U} = \langle -1, -\alpha, \alpha, 1 \rangle$ (см. рис. 5.3.2, в), для которого

$$\bar{u}^2 = 1/2(1 + \alpha^2); \quad \bar{u}^4 = 1/2(1 + \alpha^4)$$

и определитель равен

$$|\Phi| = 1/8(1 - \alpha^2)(1 - \alpha^4).$$

Его максимум очевиден при $\alpha=0$, что и дает D -оптимальный план

$$\tilde{U}^*_{D} = \langle -1, 0, 0, +1 \rangle$$

(см. рис. 5.3.2, г), для которого $|\Phi| = 1/8$. Отметим, что этот план получен из предыдущего ($N=3$) добавлением $u_4=0$.

Теперь рассмотрим несимметричный план $u_1 = -1$, $u_2 = 0$, $u_3 = u_4 = 1$. Для него получаем $|\Phi| = 1/8$, т. е. этот план

$$\tilde{U}_D^* = (-1, 0, +1, +1),$$

тоже D -оптимальный.

Анализ показывает, что для квадратичного объекта элементы D -оптимального плана при любом объеме N группируются в трех полученных выше точках $u = -1, 0, +1$. Поэтому при синтезе плана для $N > 4$ следует манипулировать только этими точками.

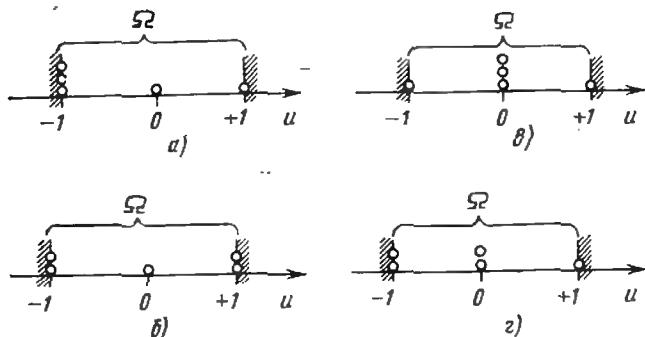


Рис. 5.3.3. К построению D -оптимальных планов при $N=5$:
а, в — неоптимальные; б, г — оптимальные планы

В общем случае, когда имеется k точек $u = -1$, q точек $u = 0$ и m точек $u = +1$ ($k + q + m = N$, $k > m$), то $\bar{u} = \bar{u}^3 = (k - m)/N$, $\bar{u}^2 = \bar{u}^4 = (k + m)/N$, и из (5.3.6) после некоторых преобразований получаем

$$|\Phi| = 4kqm/N.$$

Очевидно, что для максимизации этого выражения достаточно иметь $k = q = m = N/3$, если, разумеется, N кратно трем. В противном случае k , q и m должны различаться не более чем на единицу. Так, например, планы а и б на рис. 5.3.3 оптимальны, в и г неоптимальны.

Как видно, оптимальный план объемом $N+1$ получается из оптимального плана объемом N добавлением одного эксперимента таким образом, чтобы новый план был максимально равномерным.

Это правило можно распространить и на другие планы. Так, для $n=2$ D -оптимальный план на квадрате—

$1 \leq u_i \leq 1$ ($i=1, 2$) естественно опирается на точки, показанные на рис. 5.3.4,а. При $N=9$ эти точки образуют D -оптимальный план^{*)}. Для $N=8$ оптимальный план получается из предыдущего отбрасыванием одной из четырех точек (1, 2, 3, 4) на осях квадрата Ω (см., например, рис. 5.3.4,б). При $N=7$ следует отбросить две из указанных точек (рис. 5.3.4,в), при $N=6$ — любые три точки на осях, кроме центральной (рис. 5.3.4,г). Нарастивание числа точек оптимального плана осуществляется добавлением тех опорных точек (см. рис. 5.3.4,а), которые максимально повышают определитель $|\Phi|$.

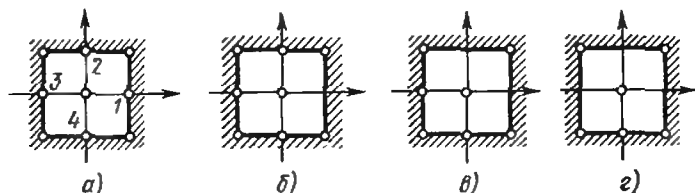


Рис. 5.3.4. D -оптимальные планы квадратичной модели ($n=2$):
а) $N=9$; б) $N=8$; в) $N=7$; г) $N=6$

Это правило имеет место и для любого другого плана любой размерности, что позволяет легко строить рекуррентно оптимальный план, последовательно добавляя опорные точки к оптимальному плану объемом $N-1$, полученному на предыдущем этапе синтеза.

5.4. Последовательное планирование экспериментов

Очень часто объем N плана заранее не определен и зависит от того, что он может дать для синтеза модели, а также от настойчивости экспериментатора. Поэтому естественно объем N заранее не оговаривать, а определять в процессе экспериментов с объектом. Это обеспечивается в ходе последовательного планирования, когда решение о последующем эксперименте принимается лишь после того, как сделан предыдущий. Кроме того, в процессе последовательного планирования экспериментатор имеет возможность изменять структуру модели, т. е. переходить от одного описания объекта к другому,

^{*)} Строго говоря, такой план не будет оптимальным [4], но его отличие от оптимального крайне незначительно.

изменять зону планирования и критерий оптимальности.

В основе последовательного планирования лежит очевидное соображение о том, что полученная в одном эксперименте информация может быть использована не только для коррекции параметров модели (как это делалось в процессах идентификации), но и для принятия других решений (например, об изменении структуры модели, прекращении экспериментов, изменении критерия или зоны планирования и т. д.).

Пусть после $(i-1)$ -го этапа планирования сделаны $i-1$ экспериментов в точках $\tilde{U}_{i-1} = \langle U_1, \dots, U_{i-1} \rangle$, т. е. реализован план $\tilde{U}_{i-1}$. Полученная при этом информация $y_1, \dots, y_{i-1}$ дала возможность синтезировать модель

$$y = f_{i-1}(U, C_{i-1}) \quad (5.4.1)$$

(например, методом наименьших квадратов или каким-либо другим методом), где f_{i-1} — структура модели, которая, по мнению экспериментатора, после $(i-1)$ -го этапа наиболее полно отражает его специфику и соответствует целям управления; C_{i-1} — значения параметров этой модели на $(i-1)$ -м этапе планирования. Располагая этой моделью, экспериментатор должен прежде всего решить, удовлетворяет его эта модель или нет. В первом случае задача синтеза модели заканчивается, во втором — ему следует оценить, покрывает ли уточненная модель затраты на производство i -го этапа эксперимента. При отрицательном решении эксперименты заканчиваются, при положительном — следует перейти к i -му этапу эксперимента. На этом этапе необходимо определить:

— новую структуру f_i модели, если экспериментатор решает ее изменить;

— новую зону Ω_i планирования эксперимента, если в этом имеется необходимость и возможность;

— новый критерий планирования K_i , если его необходимо сменить.

Если же изменений не происходит, то $f_i = f_{i-1}$, $\Omega_i = \Omega_{i-1}$, $K_i = K_{i-1}$.

Как известно, численное значение критерия K зависит от структуры модели, т. е. от выбранной системы функций $\{\varphi_j(U)\}$ ($j=1, \dots, k$) представления модели

$$y = f(U, C) = \sum_{j=1}^k c_j \varphi_j(U) \quad (5.4.2)$$

и от плана эксперимента $\tilde{U}$ (5.1.3). При последовательном планировании на i -м этапе планируется лишь один i -й эксперимент, который производится в состоянии U_i , т. е. планом является

$$\tilde{U}_i = \langle \tilde{U}_{i-1}, U_i \rangle, \quad (5.4.3)$$

где $\tilde{U}_{i-1}$ — уже реализованный к i -му этапу план.

Положение i -го эксперимента, т. е. U_i , определяется в результате решения следующей очевидной оптимизационной задачи:

$$K_i(f_i, \tilde{U}_{i-1}, U_i) \rightarrow \min_{U_i \in \Omega_i} \Rightarrow U_i^*. \quad (5.4.4)$$

Эту задачу ввиду ее сложности следует решать одним из поисковых методов оптимизации, рассмотренных в приложении.

В состоянии U_i^* ставится эксперимент, определяется выход y_i объекта и т. д., т. е. на следующем $(i+1)$ -м этапе планирования снова определяются (или сохраняются) структура модели, зона планирования и критерий. Естественно, что можно планировать не один, а несколько экспериментов, т. е. вместо U_i в выражение (5.4.4) подставить $U_i, \dots, U_{i+k} \in \Omega_i$, с помощью которых и минимизируется критерий K_i .

Так учитываются все предыдущие эксперименты $\tilde{U}_{i-1}$ и вносятся изменения, связанные с уточнением или сменой критериев, структур и т. д.

Последовательное планирование, таким образом, представляет собой очень гибкий инструмент, обеспечивающий экспериментатору возможность оперативно менять ход эксперимента, в конечном счете получить требуемую модель объекта.

5.5. Дискриминирующие планы

Дискриминирующими обычно называют планы, позволяющие выбрать одну из нескольких гипотез о структуре связи входа и выхода объекта, т. е. о структуре модели. Пусть имеется p конкурирующих структур (моделей)

$$y = f_j(U, C_j), \quad j = 1, \dots, p, \quad (5.5.1)$$

где C_j — вектор параметров j -й модели:

$$C_j = (c_{1j}, \dots, c_{k_jj}).$$

Число параметров k_j , вообще говоря, может изменяться от одной модели к другой.

Пусть имеется какой-то план, состоящий из N экспериментов, $\tilde{U} = (U_1, \dots, U_N)$, и наблюдения выхода объекта при реализации этого плана $y_1, \dots, y_N$. Тогда выбор лучшей структуры модели сводится к определению результирующей невязки каждой модели

$$Q_l = \sum_{i=1}^N [f_l(U_i, C_l) - y_i]^2, \quad l = 1, \dots, p. \quad (5.5.2)$$

Выбирают ту структуру, которая обеспечивает наименьшую невязку. Это означает, что лучшей структурой будет f_l , если

$$Q_l = \min_{j=1, \dots, p} Q_j. \quad (5.5.3)$$

План $\tilde{U}^*$, который позволяет осуществить эту процедуру наилучшим образом, назовем оптимальным дискриминирующим. Заметим, что решение о выборе структуры с минимальной невязкой (5.5.3) может быть ошибочным, так как результаты дискриминирующих экспериментов (как и всяких реальных) имеют стохастический характер. Поэтому задача синтеза оптимального дискриминирующего плана сводится к тому, чтобы сделать вероятность этой ошибки минимальной, т. е. принятое решение о выборе наилучшей структуры должно иметь наибольшую достоверность. Формализуем сказанное.

Пусть β — вероятность ошибочного решения, т. е.

$$\beta = \text{Вер}[f^* \neq f_l | Q_{\min} = Q_l], \quad (5.5.4)$$

где f^* — наилучшая структура; $Q_{\min}$ — наименьшая из (5.5.2) невязка. Эта вероятность зависит от дискриминирующего плана $\tilde{U}$: $\beta = \beta(\tilde{U})$. Тогда оптимальным дискриминирующим планом $\tilde{U}^*$ естественно назвать такой план, который минимизирует эту вероятность. Для этого необходимо решить задачу

$$\beta(\tilde{U}) \rightarrow \min \Rightarrow \tilde{U}^*, \quad \tilde{U} \in \Omega \quad (5.5.5)$$

Обсудим способы решения этой задачи. Рассмотрим случай, когда значения векторов параметров C_j ($j = 1, \dots, p$) каждой структуры известны заранее и поэто-

му могут не учитываться, т. е. $\hat{f}_j(\mathbf{U}, \mathbf{C}_i) = f_j(\mathbf{U})$. Тогда отыскание дискриминирующего плана сводится к определению положения N экспериментов $\mathbf{U}_i$ ($i=1, \dots, N$) и вычислению суммарных невязок в этих точках по формуле (5.5.2).

Пусть в частном случае ($p=2$) имеются две модели f_1 и f_2 (рис. 5.5.1). Очевидно, что оптимальным здесь следует считать дискриминирующий план, расположенный в точке, где обе модели больше всего различаются.

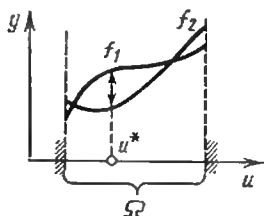


Рис. 5.5.1. К дискриминации двух одномерных моделей

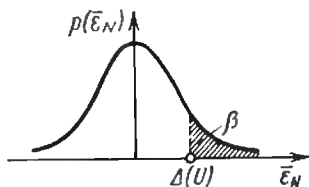


Рис. 5.5.2. К определению вероятности принятия ошибочного решения при дискриминации для двух моделей

Как легко видеть, выбор такого плана эквивалентен решению задачи (5.5.5) (ниже эта эквивалентность будет показана строго). Таким образом, для отыскания единственной точки оптимального дискриминирующего плана достаточно решить задачу

$$[f_1(\mathbf{U}) - f_2(\mathbf{U})]^2 \rightarrow \max_{\mathbf{U} \in \mathbf{U}^*}. \quad (5.5.6)$$

Решение ее — точка $\mathbf{U}^*$ — и определяет искомый план: все N экспериментов следует располагать в этой точке, т. е. $\mathbf{U}_i = \mathbf{U}^*$ ($i=1, \dots, N$).

Рассмотрим для определенности случай, когда одна из конкурирующих структур имеет место в действительности (пусть это будет f_1 , т. е. $f^0 = f_1$). Выход объекта имеет вид

$$y = f^0(\mathbf{U}) = f_1(\mathbf{U}) + \varepsilon, \quad (5.5.7)$$

где ε — случайная помеха эксперимента, которую предполагаем центрированной и нормальной, т. е. с нормальным законом распределения, нулевым ожиданием и дисперсией σ^2 . Поэтому результаты всех N экспериментов y_i ($i=1, \dots, N$) будут группироваться вокруг зна-

чения $\hat{f}_1(\mathbf{U})$ в соответствии с распределением помехи ε . Определим значения невязок Q_j ($j=1, 2$). С учетом (5.5.7) получаем

$$Q_1 = \sum_{i=1}^N \varepsilon_i^2 \quad (5.5.8)$$

$$Q_2 = \sum_{i=1}^N [f_2(\mathbf{U}) - f_1(\mathbf{U}) - \varepsilon_i]^2.$$

Пусть $\Delta(\mathbf{U})$ — разность выходов моделей в точке $\mathbf{U}$:

$$\Delta(\mathbf{U}) = f_2(\mathbf{U}) - f_1(\mathbf{U}). \quad (5.5.9)$$

Тогда после очевидных преобразований получаем

$$Q_2 = N\Delta^2(\mathbf{U}) - 2\Delta(\mathbf{U}) \sum_{i=1}^N \varepsilon_i + \sum_{i=1}^N \varepsilon_i^2.$$

Как легко заметить, знак разности

$$\delta(\mathbf{U}) = Q_2(\mathbf{U}) - Q_1(\mathbf{U}) = N\Delta^2(\mathbf{U}) - 2\Delta(\mathbf{U}) \sum_{i=1}^N \varepsilon_i$$

определяет выбор одной из моделей. Действительно, если эта разность положительна, то следует выбирать f_1 , а если отрицательна, то f_2 (напомним, что числа Q_i неотрицательны по определению). Таким образом,

$$f_i = \begin{cases} f_1 & \text{при } \delta(\mathbf{U}) > 0, \\ f_2 & \text{при } \delta(\mathbf{U}) < 0. \end{cases} \quad (5.5.10)$$

Это и есть правило решения задачи (5.5.6), а следовательно, и (5.5.5).

Определим вероятность ошибки, т. е. вероятность принятия неправильного решения при $\Delta(\mathbf{U}) > 0$:

$$\beta = \text{Вер} \left[\frac{1}{N} \sum_{i=1}^N \varepsilon_i > \frac{1}{2} \Delta(\mathbf{U}) \right], \quad (5.5.11)$$

т. е. вероятность того, что случайная величина

$$\bar{\varepsilon}_N = \frac{1}{N} \sum_{i=1}^N \varepsilon_i$$

превысит $1/2 \Delta(\mathbf{U})$. Зная вероятностные свойства $\bar{\varepsilon}_N$:

$$M(\bar{\varepsilon}_N) = 0; \quad D(\bar{\varepsilon}_N) = \sigma^2/N,$$

легко определить искомую вероятность. Она равна заштрихованной площади под кривой распределения случайной величины ε_N (рис. 5.5.2). Для нормального закона распределения ε_N

$$\beta = \frac{1}{2} \left[1 - \Phi \left(\frac{\Delta(\mathbf{U}) \sqrt{N}}{2 \sqrt{2} \sigma} \right) \right], \quad (5.5.12)$$

где $\Phi(\cdot)$ — функция Лапласа, которая является монотонно возрастающей. Это свойство функции $\Phi(\cdot)$ и определяет выбор искомого плана, который для минимизации вероятности ошибки должен максимизировать величину $\Delta(\mathbf{U})$, т. е. следует решать задачу (5.5.6). Если подставить полученную точку $\mathbf{U}^*$ в выражение (5.5.12), то получим вероятность ошибки при оптимальном плане. С другой стороны, вероятность ошибки при таком решении можно уменьшить, увеличив число экспериментов N в точке $\mathbf{U}^*$. Таким образом, можно планировать эксперименты, которые обеспечивают выбор модели с любой наперед заданной степенью достоверности.

Однако в реальных задачах параметры $\mathbf{C}_j$ ($j=1, \dots, p$) моделей (5.5.1) неизвестны, и поэтому одновременно с дискриминацией следует производить оценку этих параметров. Сделать это можно путем последовательного планирования экспериментов следующим образом.

Пусть для простоты $p=2$, т. е., как и выше, решается задача дискриминации двух моделей $f_1(\mathbf{U}, \mathbf{C}_1)$ и $f_2(\mathbf{U}, \mathbf{C}_2)$, где векторы параметров $\mathbf{C}_1$ и $\mathbf{C}_2$ неизвестны. На $(i+1)$ -м этапе дискриминации имеем параметры $\mathbf{C}_{1i}$ и $\mathbf{C}_{2i}$, для которых решаем задачу синтеза плана дискриминации

$$|f_1(\mathbf{U}, \mathbf{C}_{1i}) - f_2(\mathbf{U}, \mathbf{C}_{2i})| \rightarrow \max_{\mathbf{U} \in \Omega} \Rightarrow \mathbf{U}_{i+1}. \quad (5.5.13)$$

Определив в эксперименте значение выхода объекта

$$y_{i+1} = f^0(\mathbf{U}_{i+1}),$$

можно откорректировать параметры, воспользовавшись известным приемом адаптивной идентификации (см. § 4.2):

$$\mathbf{C}_{j, i+1} = \mathbf{C}_{j, i} + \Delta \mathbf{C}_{j, i+1}, \quad j=1, 2, \quad (5.5.14)$$

где

$$\Delta \mathbf{C}_{j, i+1} = -2\alpha_{i+1} q_{i+1}'(\mathbf{C}_{j, i}) \nabla_{\mathbf{C}} f(\mathbf{U}_{i+1}, \mathbf{C}_{j, i}),$$

а входящие в эту формулу выражения определены выше в (4.2.14) и (4.2.16). Полученные новые значения параметров дают возможность аналогично определить новую точку дискриминации U_{i+2} и т. д.

Решение о выборе модели принимается тогда, когда имеется уверенность, что параметры моделей определены правильно. В этом случае при

$$\sum_{i=1}^N [f_1(U_i, C_{1N}) - y_i]^2 < \sum_{i=1}^N [f_2(U_i, C_{2N}) - y_i]^2$$

выбирается первая модель, а при обратном неравенстве — вторая. Вероятность ошибочного решения здесь оценивается по приближенной формуле

$$\beta \approx 1 - \prod_{i=1}^N (1 - \beta_i), \quad (5.5.15)$$

где $\beta_i = \frac{1}{2} \left[1 - \Phi \left(\frac{\Delta_i}{2\sqrt{2}\sigma} \right) \right]$,

$$\Delta_i = |f_1(U_i, C_{1N}) - f_2(U_i, C_{1N})|.$$

Несовпадение формул (5.5.12) и (5.5.15) при $U_i = U$ связано с приближенностью оценки (5.5.15). При $N=1$ они совпадают.

Задачи планирования дискриминирующих экспериментов имеет двойственный характер. Здесь надо одновременно добиться двух целей — определить параметры модели и выбрать наилучшую. Противоречивость этих требований очевидна. Действительно, для эффективности определения параметров следует точки плана располагать на границе области планирования, а для дискриминации — в точках наибольшего различия моделей. Необходимо компромисс, один из которых и изложен выше.

Таким образом, этапы синтеза модели (это этапы 3—5 на схеме рис. В.4.1) заканчиваются созданием модели F объекта управления, которая необходима для синтеза управления всяким сложным объектом.

ЛИТЕРАТУРНЫЙ КОММЕНТАРИЙ К ГЛАВЕ 5

Планирование экспериментов является разделом прикладной математической статистики и поэтому нелегко популяризуется.

Для начального ознакомления с основными идеями планирования экспериментов можно порекомендовать очень интересную и мастерски написанную научно-популярную книгу [1] и менее популярную, но весьма насыщенную [2].

Термин «планирование экспериментов» имеет два содержания. Во-первых, так называют способы организации эксперимента на реальном объекте для определения его модели (в этом смысле этот термин и был использован в данной книге). Однако есть и другой смысл — планирование экстремальных экспериментов, т. е. экспериментальное определение оптимальных условий функционирования объекта. По сути дела, это последнее направление в планировании эксперимента является разделом теории управления объектами типа «черного ящика», т. е. объектами, не имеющими математического описания. С этим важным для практики направлением можно познакомиться по очень доступной и весьма содержательной книге [3].

Но задачу экспериментального определения неизвестных параметров объекта решают иначе, так как известна структура модели объекта. По такого рода планированию эксперимента имеется ряд очень содержательных, хотя и не вполне доступных книг (для их чтения необходимо знать теорию вероятностей и линейную алгебру). Отметим среди них наиболее полные [4–6], а также [7, 8].

Всего было предложено и использовано более 100 различных критериев планирования (по литературным данным). Они четко подразделяются на два класса. К первому относятся критерии, определенные на дисперсионной матрице. Значения этих критериев можно вычислять без экспериментов. Некоторые из них рассмотрены в этой главе. Ко второму классу относятся критерии, связанные с величиной невязки выходов объекта и его модели, т. е. с адекватностью модели. Для определения значений этих критериев необходимо иметь результаты экспериментов с объектом.

СПИСОК ЛИТЕРАТУРЫ

1. Хургин Я. И. Да, нет или может быть... — М.: Наука, 1977.
2. Налимов В. В. Теория эксперимента. — М.: Наука, 1971.
3. Адлер Ю. П., Маркова Е. В., Грановский Ю. В. Планирование эксперимента при поиске оптимальных условий. — М.: Наука, 1976.
4. Федоров В. В. Теория оптимального эксперимента: планирование регрессионных экспериментов. — М.: Наука, 1971.
5. Круг Г. К., Сосулин Ю. А., Фатуев В. А. Планирование эксперимента в задачах идентификации и экстраполяции. — М.: Наука, 1977.
6. Денисов В. И. Математическое обеспечение системы ЭВМ — экспериментатор: регрессионный и дисперсионный анализы. — М.: Наука, 1977.
7. Планирование эксперимента в исследовании технологических процессов: Пер. с англ./ Под ред. Э. К. Лецкого. — М.: Мир, 1977.
8. Химмельблау Д. Анализ процессов статистическими методами: Пер. с англ. — М.: Мир, 1973.

СИНТЕЗ УПРАВЛЕНИЯ (ПРИНЯТИЕ РЕШЕНИЯ)

Для реализации этого этапа необходимо иметь модель объекта $F(X, U)$, информацию о состоянии среды X и цель управления Z^* . Процедуру определения управления U^* , решающего поставленную задачу в сложившейся ситуации, называют алгоритмом управления. Однако прежде, чем говорить об этом алгоритме, сформулируем задачу.

6.1. Постановка задачи

Ситуация, которая складывается в процессе управления, характеризуется тройкой

$$(X, Y, Z^*). \quad (6.1.1)$$

Здесь с помощью управления можно изменять лишь состояние Y объекта, модель которого

$$Y = F(X, U) \quad (6.1.2)$$

имеется в нашем распоряжении. Достижение целей управления Z^* , как показано в гл. 1, эквивалентно выполнению условий (1.3.8), которые при учете того, что $S = \langle X, Y \rangle$ (см. (1.3.12)), сводятся к выполнению следующих целевых соотношений:

$$Z^*: \begin{cases} \psi_i(X, Y) = a_i & (i = 1, \dots, k_1), \\ \psi_j(X, Y) = b_j & (j = 1, \dots, k_2), \\ \psi_l(X, Y) \rightarrow \min & (l = 1, \dots, k_3), \end{cases} \quad (6.1.3)$$

где функции ψ_i , ψ_j и ψ_l , а также числа a_i и b_j должны быть заданы на стадии формулировки целей управления (см. гл. 1). Запишем эти целевые условия в канонической векторной форме:

$$Z^*: \begin{cases} H(X, Y) \geq 0, \\ G(X, Y) = 0, \\ Q(X, Y) \rightarrow \min, \end{cases} \quad (6.1.4)$$

где

$$\begin{aligned} G(X, Y) &= (g_1(X, Y), \dots, g_{k_1}(X, Y)); \\ H(X, Y) &= (h_1(X, Y), \dots, h_{k_2}(X, Y)); \\ Q(X, Y) &= (q_1(X, Y), \dots, q_{k_3}(X, Y)); \end{aligned} \quad (6.1.5)$$

причем

$$\begin{aligned} g_i(X, Y) &= \psi_i(X, Y) - a_i, \quad i=1, \dots, k_1; \\ h_j(X, Y) &= \psi_j(X, Y) - b_j, \quad j=1, \dots, k_2; \\ q_l(X, Y) &= \psi_l(X, Y), \quad l=1, \dots, k_3. \end{aligned}$$

Введенные функции удобно называть следующим образом $G(\dots)$ — функция ограничений типа равенств, $H(\dots)$ — функция ограничений типа неравенств и $Q(\dots)$ — функция качества.

Реализовать условия (6.1.4) можно только с помощью соответствующего изменения состояния Y объекта управления. Это означает, что их выполнение возможно только при соответствующем выборе управления U (состояние X среды изменяется независимо от нас). Эти очевидные соображения приводят к следующей экстремальной задаче:

$$Q(X, Y) \rightarrow \min_{U \in \Omega} \quad (6.1.6)$$

где

$$\Omega: \begin{cases} H(X, Y) \geq 0, \\ G(X, Y) = 0, \\ Y = F(X, U), \end{cases}$$

решение которой U^* и является оптимальным управлением. Здесь использована модель F (6.1.2) и учтен ресурс, выделяемый на управление, — он выражен в виде соответствующих неравенств и равенств в Ω .

Рассмотрим задачу (6.1.6) подробнее. Она требует минимизации Q путем подбора соответствующего управления U , которое должно, кроме того, удовлетворить условиям Ω . Создается впечатление, что мы как бы про-ранжировали наши цели, причем основными названы экстремальные цели Q , а неэкстремальные «загнаны» в ограничения Ω . Однако это лишь кажущееся ранжирование. В действительности основным является выпол-

нение ограничений Ω , а экстремальные цели Q являются вторичными: они достигаются лишь при условии выполнения неэкстремальных целей, т. е. ограничений Ω .

Таким образом, задача синтеза управления (6.1.6) характеризуется двойкой $\langle Q, \Omega \rangle$ — экстремальными и неэкстремальными целями, причем достижению первых должно предшествовать выполнение вторых. Это естественное требование широко используется нами в обыденной жизни. Действительно, мы прежде всего заботимся о том, чтобы не впасть в противоречие с ограничениями Ω . Это, например, правила уличного движения, моральный и уголовный кодекс, правила общежития, противопожарной безопасности и т. д. И лишь при соблюдении этих естественных ограничений мы выполняем наши экстремальные цели Q (жизненный успех и удовольствия, счастье, признание окружающих и т. д.). А тот, кто слишком увлекается достижением экстремальных целей Q в ущерб ограничениям Ω , рискует вступить в противоречие с законами, как моральными, так и юридическими. Рассматриваемую задачу (6.1.6) удобно преобразовать и записать в виде

$$Q(X, F(X, U)) \rightarrow \min_{U \in \Omega} \quad (6.1.7)$$

где

$$\Omega: \begin{cases} (X, F(X, U)) = 0, \\ H(X, F(X, U)) \geq 0. \end{cases}$$

В зависимости от вида модели F , т. е. от того, является ли F функцией или оператором, получаем различные задачи, которые следует решать разными методами.

Рассмотрим сначала статический объект управления, для которого модель F является функцией. Задача синтеза управления в этом случае заключается в минимизации k_3 функций $q_l(X, U)$ ($l=1, \dots, k_3$) путем изменения q параметров $u_1, \dots, u_q$, удовлетворяющих ограничениям Ω , наложенным на U . Эти ограничения связаны с выполнением условий неотрицательности и равенства нулю заданных функций управления U . Состояние X среды при этом предполагается, естественно, известным.

Примером является задача планирования производства. В этом случае U — план производства; X — поставки сырья; G — требования

к номенклатуре производимых изделий, соблюдению норм безопасности в процессе производства и т. д.; **Н** — требования к качеству продукции (не хуже, чем), различного рода производственные нормативы (от... и до...) и ограничения (не более, чем...). В качестве экстремальных целей **Q** при планировании производства могут быть, например, производительность труда, себестоимость продукции, объем ее выпуска, отклонение от плана (его надо минимизировать) и т. д. Моделью **F** производства здесь являются рассуждения, позволяющие определить состояние **Y** производства при известном состоянии среды **X** и плане производства **U**. Эти рассуждения обычно носят имитационный характер (см. § 3.5).

Таким образом, задача *математического программирования*, возникающая при управлении статическими объектами, отличается тем, что искомое управление представляет собой набор параметров $u_1, \dots, u_q$, а **Q**, **G** и **H** являются векторными функциями управления **U**.

Теперь рассмотрим динамический объект, для которого **F** — оператор. В этом случае управление **U** представляет собой векторную функцию времени $U(t)$, а q_i , h_i и g_j являются функционалами, а задачу (6.1.7) называют *вариационной*.

Примером является задача о таксисте, которому нужно с минимальным временем пройти от одного нерегулируемого перекрестка до другого (нерегулируемость требует остановки перед перекрестком). Уравнение движения такси, т. е. оператор f модели, имеет вид

$$f: \frac{d^2 y}{dt^2} + c_1 \frac{dy}{dt} = c_2 u(t),$$

где y — пройденный путь, причем $y(0) = \dot{y}(0) = 0$, т. е. старт берется в нулевой момент времени с нулевой скоростью; c_1 и c_2 — параметры торможения и разгона автомашины, они определяются на стадии идентификации (или планирования эксперимента); $u(t)$ — скалярное управление ($q=1$) ускорением движения, которое осуществляется педалью (акселератором) и тормозом. Оно представляет собой силу, действующую на машину $a_1 \leq u(t) \leq a_2$, где $a_1 < 0$ — максимальная сила торможения (зависит от качества тормозов и состояния дорожного покрытия); $a_2 > 0$ — максимальное усилие (зависит от мощности двигателя, качества бензина и т. д.).

Пусть T — время движения от одного перекрестка до другого. Эта величина является функционалом, значение которого определяется выбранным законом управления $u(t)$. Для нахождения значения этого функционала необходимо решить уравнение модели с граничными условиями

$$y(0) = \dot{y}(0) = \dot{y}(T) = 0; \quad y(T) = L,$$

где L — расстояние между перекрестками. Полученное при этом значение функционала $T = T(u(t))$ и является экстремальной (минимизируемой) целью.

Теперь задачу (6.1.6) для рассматриваемого случая можно записать в виде

$$T(u(t)) \rightarrow \min, \\ u(t) \in \Omega$$

$$\Omega: \begin{cases} a_1 \leq u(t) \leq a_2, \\ \ddot{y} + c_1 \dot{y} = c_2 u(t), \\ y(0) = \dot{y}(0) = \dot{y}(T) = 0, \quad y(T) = L, \\ \dot{y}(t) \leq a_3. \end{cases}$$

Здесь к уже известным ограничениям добавлено ограничение по скорости ($a_3=60$ км/ч), которое предписывается правилами дорожного движения. Получаем оптимальное управление в виде функции

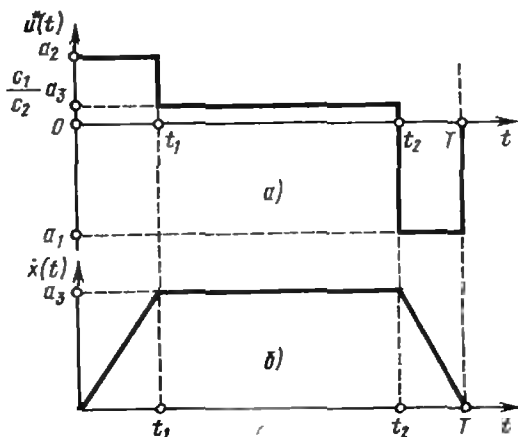


Рис. 6.1.1. Пример оптимального управления

$u^*(t)$. Решить эту задачу сложно, однако здравый смысл подсказывает оптимальное решение, которое и используется таксистами. Оно состоит в следующем: сначала «давить на всю железку», т. е. $u(t)=a_2$ до момента $\dot{y}=a_3$, затем поддерживать скорость на максимально допустимом правилами уровне a_3 (из уравнения модели получаем для этого режима $u(t)=(c_1/c_2)a_3$ и перед перекрестком резко тормозить ($u(t)=a_1$) до полной остановки, т. е. $\dot{y}=0$).

На рис. 6.1.1 показан график этого оптимального управления $u^*(t)$ (a) и изменение скорости при этом (б).

Как видно, задача синтеза оптимального управления $u^*(t)$ сводится к определению лишь двух моментов времени t_1 (конца разгона) и t_2 (начала торможения). Заметим, что влияние среды X (в данном случае качества дорожного покрытия) может изменить длину тормозного пути, т. е. $a_1=a_1(X)$, и максимальную скорость движения в условиях X , т. е. $a_3=a_3(X)$.

Полученное решение будет оптимальным для таксиста, но явно неприемлемо для пассажира, которому придется испытывать огромные перегрузки (особенно при торможении). Поэтому в интересах пассажира необходимо снизить перегрузки до приемлемого уровня $a^*_1 > a_1$ (напомним, что это отрицательные величины). Здесь значение a^*_1 определяется комфортными условиями для пассажира.

Управление динамическим объектом, таким образом, сводится к решению вариационной задачи (6.1.6), в которой искомое управление представляет собой функцию времени $U=U(t)$, а Q , G и H являются, вообще говоря, заданными функционалами управления $U(t)$.

Очевидно, что эта задача значительно сложнее задачи математического программирования. Поэтому решение вариационной задачи с произвольными функционалами обычно сводят к решению задач математического программирования. Для этого достаточно параметризовать функцию управления $U(t)$.

Сделать это можно двумя способами:

1) путем разложения $U(t)$ в ряд по заданной системе функций $\{\Phi_i(t)\}$:

$$U(t) = \sum_{i=1}^k c_i \Phi_i(t),$$

что приводит к задаче отыскания параметров $c_1, \dots, c_k$. Этот подход называют параметрическим;

2) представление функции $U(t)$ ее дискретами $C_1, \dots, C_k$ в заданные моменты времени, где $C_i = U(t_i)$ ($i=1, \dots, k$) — новые искомые параметры управления; этот подход обычно называют непараметрическим, хотя искомые параметры здесь тоже имеются.

В обоих случаях, подставляя эти выражения в (6.1.7), получаем после очевидных преобразований задачу математического программирования вида

$$\begin{aligned} Q(X, U) \rightarrow \min_{U \in \Omega} \\ \Omega: \begin{cases} G(X, U) = 0, \\ H(X, U) \geq 0, \end{cases} \end{aligned} \quad (6.1.8)$$

где $U=(C_1, \dots, C_k)$ — вектор искомых параметров, а Q , G и H — определенные векторные функции этих параметров.

Таким образом, в конечном счете решение любой задачи синтеза управления сводится к решению задачи математического программирования. Рассмотрим различные типы задач такого рода.

6.2. Классификация задач математического программирования

Вид и число функций Q , G и H определяют различные типы задач математического программирования, а следовательно, и методы их решения (для этого и вводится классификация). В зависимости от числа k_3 минимизируемых функций q_l ($l=1, \dots, k_3$) различают *однокритериальные* (при $k_3=1$) и *многокритериальные* (при $k_3>1$) задачи. В последнем случае их иногда называют задачами векторной оптимизации (задачами оптимизации с векторным критерием). Спецификой многокритериальной задачи математического программирования является необходимость введения дополнительной информации об объекте и критериях, без чего эта задача не имеет решения. Методы решения многокритериальных задач рассмотрены в приложении.

Далее будем рассматривать лишь однокритериальные задачи ($k_3=1$), для которых Q — скалярная функция, т. е. $Q=Q$.

В зависимости от вида и характера функций Q , G и H различают следующие задачи математического программирования.

1. Если Q , G и H линейны по U , то имеем задачу *линейного* программирования:

$$Q(U) = \sum_{l=1}^q a_l u_l \rightarrow \min, \quad U \in \Omega \quad (6.2.1)$$

где $\Omega: \begin{cases} \sum_{k=1}^q a_{ik} u_k = c_i, & i = 1, \dots, k_1 < q, \\ \sum_{k=1}^q b_{jk} u_k \geq d_j, & j = 1, \dots, k_2. \end{cases}$

Здесь параметры a_l , a_{ik} , b_{jk} , d_j и c_i определяются факторами среды X и объекта F .

Процесс решения задачи линейного программирования естественно ориентировать на линейность выражений (6.2.1), что реализуется так называемым симплексным методом, который применим только для решения задач такого рода.

Задача об оптимальном коктейле, рассмотренная выше (см. § В.5), являлась задачей линейного программи-

рования. Другим примером такого рода задач является задача об оптимальной смеси. Рассмотрим ее.

Пример. Из q видов исходного сырья $A_1, A_2, \dots, A_q$, каждый из которых состоит из n компонентов

$$A_k = (a_{1k}, a_{2k}, \dots, a_{nk}), \quad k=1, \dots, q$$

(здесь a_{ik} — относительное содержание i -го компонента в k -м сырье), необходимо составить смесь заданного состава: $c_1, \dots, c_n$, где c_i — относительное содержание i -го компонента в смеси.

Смесь характеризуется вектором состава $U = (u_1, \dots, u_q)$, где u_k — относительное количество k -го сырья $\left(\sum_{k=1}^q u_k = 1 \right)$. Стои-

мость сырья определяется ценами: $a_1, \dots, a_q$, где a_l — стоимость единицы l -го сырья ($l=1, \dots, q$). Стоимость единицы смеси при этом должна быть минимальной.

Задача, таким образом, сводится к минимизации линейной функции стоимости смеси:

$$Q(U) = \sum_{l=1}^q a_l u_l \rightarrow \min, \quad U \in \Omega$$

где множество Ω допустимых значений состава U определяется требованиями, налагаемыми на состав смеси:

$$\Omega: \begin{cases} \sum_{k=1}^q a_{lk} u_k = c_l, \quad l=1, \dots, n, \\ u_k \geq 0, \quad k=1, \dots, q. \end{cases}$$

Здесь последнее неравенство $u_k \geq 0$ выражает очевидное требование, связанное с тем, что количество сырья в смеси не может быть отрицательным. Как видно, ограничения Ω здесь состоят из n равенств (Г) и q неравенств (Н), причем $n < q$ (при $n=q$ задача вырождается в неэкстремальную). Это типичная задача линейного программирования.

Методы решения задач линейного программирования рассмотрены в § 6.4.

2. Другой вид задач математического программирования — задачи выпуклого программирования, они имеют более общий характер и формулируются следующим образом.

Необходимо минимизировать выпуклую вниз функцию

$$Q(U) \rightarrow \min, \quad U \in \Omega \quad (6.2.2)$$

где область Ω допустимых состояний задается системой неравенств

$$\Omega : h_i(\mathbf{U}) \geq 0, i=1, \dots, k_2. \quad (6.2.3)$$

Здесь $h_i(\mathbf{U})$ — выпуклые вниз функции, что гарантирует выпуклость области Ω .

Заметим, что функция $f(\mathbf{X})$ называется выпуклой вниз, если отрезок, соединяющий две любые точки этой функции, лежит выше ее значений (рис. 6.2.1). Формально это определение можно записать в следующем виде.

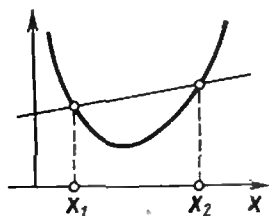


Рис. 6.2.1. К определению выпуклой функции

Если для любой точки $\mathbf{X}$, лежащей между точками $\mathbf{X}_1$ и $\mathbf{X}_2$, т. е. при

$$\mathbf{X} = \lambda \mathbf{X}_1 + (1-\lambda) \mathbf{X}_2, 0 \leq \lambda \leq 1, \quad (6.2.4)$$

выполняется неравенство

$$f(\mathbf{X}) \leq \lambda f(\mathbf{X}_1) + (1-\lambda) f(\mathbf{X}_2), \quad (6.2.5)$$

то эта функция выпукла вниз. Как легко убедиться, правая часть неравенства (6.2.5) является линейной аппроксимацией функции $f(\mathbf{X})$ по двум ее значениям в точках $\mathbf{X}_1$ и $\mathbf{X}_2$ (см. прямую на рис. 6.2.1).

Выпуклой областью Ω принято называть множество точек, обладающих следующим свойством. Если взять две любые точки этого множества, то отрезок прямой, соединяющий эти точки, тоже должен принадлежать этому множеству. Формально: если $\mathbf{X}_1, \mathbf{X}_2 \in \Omega$, то $\mathbf{X} \in \Omega$, где точка $\mathbf{X}$ определена (6.2.4) как точка между $\mathbf{X}_1$ и $\mathbf{X}_2$.

Выпуклые функции представляют собой довольно широкий класс функций и сравнительно хорошо могут описывать задачи (6.1.7) синтеза управления.

Существенной особенностью задачи выпуклого программирования является ее локальность, т. е. наличие одного экстремума. Это означает, что при решении задач выпуклого программирования можно довольствоваться отысканием локального экстремума. Свойства

выпуклости минимизируемой функции Q и области Ω используются при решении задач выпуклого программирования. Заметим, что задачи линейного программирования являются частным случаем выпуклого программирования. В этом случае легко убедиться, так как любая линейная функция удовлетворяет условиям выпуклости. Метод решения задач выпуклого программирования описан в § 6.4.

3. Требования выпуклости далеко не всегда выполняются в задачах синтеза управления. В общем случае, когда Q , G , H являются произвольными нелинейными функциями, решается задача *нелинейного программирования*.

Это весьма общий вид математического программирования. Здесь, вообще говоря, нельзя гарантировать совпадение локального и глобального экстремумов, что часто создает серьезные трудности при решении такого рода задач, которые в этом случае относятся к классу многоэкстремальных. Однако даже при одноэкстремальности задачи специальных математических методов решения задач нелинейного программирования не существует. Все применяемые для этого методы являются поисковыми методами оптимизации (некоторые из них рассмотрены в приложении).

Типичным примером задачи нелинейного программирования могут служить задачи оптимального проектирования. При проектировании новых схем, конструкции и систем всегда определена экстремальная цель, например эффективность, надежность, быстродействие, масса конструкции и т. д. Оптимальное проектирование представляет собой процесс определения параметров $u_1, \dots, u_q$ конструкции, которые экстремизируют эту цель, но не произвольно, а при соблюдении определенных ограничений. Вид ограничений связан со строгим выполнением ряда требований. Например, необходимо использовать элементы конструкции определенного вида. Это — ограничение типа равенств. Другие ограничения имеют характер неравенств. Например, температура и возникающие напряжения не должны превышать заданных пределов норм и т. д. Таким образом, задачи оптимального проектирования являются типичными задачами нелинейного программирования, так как ограничения и функция качества имеют обычно ярко выраженный нелинейный характер.

4. Очень часто управление U является дискретным, т. е. множество допустимых управлений Ω конечно и дискретно. Например, если компоненты вектора управления u_i ($i=1, \dots, q$) могут принимать только целочисленные значения, в этом случае решается задача *целочисленного программирования*.

Пример. Задачей целочисленного программирования может служить задача управления системой, состоящей из взаимодействующих агрегатов, каждый i -й агрегат может либо работать ($u_i=1$), либо не работать ($u_i=0$). Эффективность работы этой системы $Q(U)$ должна быть максимизирована при заданном энергетическом ресурсе R :

$$Q(U) \rightarrow \max_{U \in \Omega},$$

$$\Omega: \begin{cases} \sum_{i=1}^q a_i u_i \leq R, \\ u_i = \{0, 1\}, \quad i = 1, \dots, q. \end{cases}$$

где a_i — затраты на функционирование i -го агрегата.

Если в простейшем случае агрегаты не взаимодействуют, то

$$Q(U) = \sum_{i=1}^q b_i u_i,$$

где b_i — эффективность работы i -го агрегата.

В результате получаем *целочисленную задачу линейного программирования*.

Решение задач такого рода иногда представляет серьезные трудности. Они будут рассмотрены в § 6.4.

5. Если Q , G и H являются вероятностными характеристиками, то решается *задача стохастического программирования*.

Напомним, что наиболее распространенными вероятностными характеристиками являются:

вероятность

$$P(X \in S) = \int_S p(X) dX, \quad (6.2.6)$$

математическое ожидание

$$M(X) = \int X p(X) dX, \quad (6.2.7)$$

дисперсия

$$D(X) = \int [X - M(X)]^2 p(X) dX \quad (6.2.8)$$

и т. д., где $p(X)$ — плотность распределения случайного фактора X , а интеграл многократный: он имеет кратность, равную размерности вектора X .

Примером задачи стохастического программирования является осреднение исходной задачи (6.1.8) по случайному состоянию X среды, т. е. когда эту задачу необходимо решать в среднем. Тогда (6.1.8) преобразуется к виду

$$\begin{aligned} \bar{Q}(U) \rightarrow \min_{U \in \Omega}, \\ \Omega: \begin{cases} \bar{G}(U) = 0, \\ \bar{H}(U) \geq 0, \end{cases} \end{aligned}$$

где

$$\begin{aligned} \bar{Q}(U) &= \int Q(X, U) p(X) dX; \quad \bar{G}(U) = \int G(X, U) p(X) dX \\ \bar{H}(U) &= \int H(X, U) p(X) dX, \end{aligned}$$

$p(X)$ — известная плотность распределения состояния X среды. Эта функция n переменных $x_1, \dots, x_n$ — входов объекта с естественными ограничениями вида

$$p(X) \geq 0, \quad \int p(X) dX = 1.$$

Интеграл здесь и выше является n -кратным и берется по всему пространству входов $\{X\}$.

Как видно, решение задачи стохастического программирования сводится к определению вероятностных характеристик, т. е. к вычислению многократных интегралов вида

$$J = \int f(X, U) p(X) dX, \quad (6.2.9)$$

что само по себе представляет серьезную вычислительную задачу. Однако эту трудность можно преодолеть, если воспользоваться для J его оценкой по методу Монте-Карло:

$$\hat{J}_N = \frac{1}{N} \sum_{i=1}^N f(X_i, U), \quad (6.2.10)$$

где X_i ($i=1, \dots, N$) — независимые случайные векторы, генерируемые в соответствии с заданной плотностью

распределения $p(\mathbf{X})$. Действительно, J является средним значением функции f :

$$J = M_{\mathbf{X}} f(\mathbf{X}, \mathbf{U}),$$

где $M_{\mathbf{X}}$ — математическое ожидание по $\mathbf{X}$. Но выражение (6.2.10) является оценкой этого среднего значения на базе N , поэтому может служить приближенным значением (6.2.9).

Оценка (6.2.10), полученная методом Монте-Карло, является случайной величиной и характеризуется следующими вероятностными свойствами. Ее среднее значение совпадает с J : $M(\hat{J}_N) = J$, т. е. оценка $\hat{J}_N$ несмещенная, и в пределе $\lim_{N \rightarrow \infty} \hat{J}_N = J$, т. е. оценка состоятельная,

а ее дисперсия $D(\hat{J}_N) = c/N$, где c — постоянная. Это означает, что оценка, полученная методом Монте-Карло, «хорошая», т. е. в среднем при любом N совпадает с истинной.

Эти свойства дают возможность записать выражение для связи оценки и точного значения в виде

$$\hat{J}_N = J + \varepsilon(c/N), \quad (6.2.11)$$

где $\varepsilon(c/N)$ — случайная величина с нулевым средним и дисперсией c/N .

Выражение (6.2.11) означает, что оценка, полученная методом Монте-Карло, отличается от точного ее значения на случайную величину, в среднем равную нулю, дисперсия которой уменьшается с ростом N как $1/N$. Это позволяет интерпретировать $\hat{J}_N$ как результат вычисления J со случайной помехой, которая обусловлена способом вычисления. Поэтому задачу стохастического программирования можно записать в стандартной форме (6.1.8), где, однако, на каждую из используемых функций в процессе решения налагается случайная помеха, свойства которой, вообще говоря, могут быть неизвестны. Решение подобной задачи представляет естественные трудности, которые могут быть преодолены при использовании поисковых методов типа стохастической аппроксимации (см. приложение),

6.3. Анализ задач математического программирования

Основным методом решения задач математического программирования является так называемый *рекуррентный метод*. Он заключается в построении последовательности приближенных решений задачи

$$U_0, U_1, \dots, U_N, \dots, \quad (6.3.1)$$

которая обладает следующими важными свойствами:

1) каждый последующий член этой последовательности в определенном смысле (с точки зрения решения задачи) предпочтительней предыдущего, т. е. $U_{N+1} > U_N$, где $>$ — знак предпочтения;

2) эта последовательность при больших N приближается к значению U^* , которое может рассматриваться как решение задачи. Следует отметить, что точного решения задачи математического программирования получать обычно не требуется, так как исходные данные в задаче всегда приближены. Поэтому речь может идти о приближении к рациональному решению. И лишь в частном случае U^* может быть точным решением задачи.

Рассмотрим способы формального определения указанного предпочтения. Это можно сделать различными способами. Приведем примеры введения очевидных предпочтений.

1. $U_{N+1} > U_N$, если $U_N, U_{N+1} \in \Omega$ и $Q(U_{N+1}) < Q(U_N)$, т. е. из допустимых решений (тех, которые удовлетворяют Ω) лучше то, которое дает меньшее значение функции качества $Q(U)$.

2. Если же $Q(U_{N+1}) = Q(U_N)$ и $U_{N+1}, U_N \in \Omega$, то $U_{N+1} \sim U_N$, т. е. решения эквивалентны ($\sim$ — знак эквивалентности).

3. $U_{N+1} > U_N$, если $U_{N+1} \in \Omega$, а $U_N \notin \Omega$, т. е. более предпочтительным всегда является допустимое решение.

Этот список можно было бы продолжить. Однако задание предпочтений не решает полностью задачу синтеза алгоритма решения поставленной задачи в любой ситуации, сложившейся в процессе управления. Описанные выше предпочтения, как легко заметить, описывают далеко не все ситуации, которые могут возникнуть при управлении объектом. Эту задачу выполняет алгоритм. Таким алгоритмом решения задачи математического

программирования будем называть оператор Φ , связывающий последующие решения:

$$\mathbf{U}_N = \mathbf{U}_{N-1} + \Phi(I_N, \mathbf{U}_{N-1}, Q_{N-1}, \mathbf{G}_{N-1}, \mathbf{H}_{N-1}).$$

Здесь Φ — алгоритм, предписывающий, как должно измениться управление при имеющихся значениях $\mathbf{U}_{N-1}$, $Q_{N-1} = Q(\mathbf{U}_{N-1})$, $\mathbf{G}_{N-1} = \mathbf{G}(\mathbf{U}_{N-1})$, $\mathbf{H}_{N-1} = \mathbf{H}(\mathbf{U}_{N-1})$ и полученной дополнительной информации I_N (все это и есть сложившаяся в процессе управления ситуация).

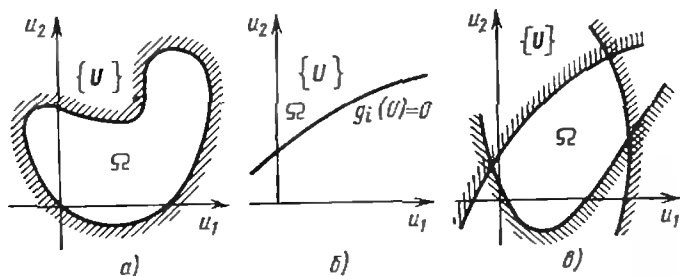


Рис. 6.3.1. Область допустимых управлений:

a — общий случай; b — ограничения типа равенств; c — образование Ω с помощью неравенств

Рассмотрим пространство решений $\{U\}$. В этом пространстве условия Ω выделяют определенную область допустимых решений $U \in \Omega$ (рис. 6.3.1, a). Размерность области Ω равна $q - k_1$, так как каждое ограничение — равенство $g_i(U) = 0$ — уменьшает размерность задачи на единицу (рис. 6.3.1, b). Ограничения типа неравенства $h_j(U) \geq 0$ очерчивают границы области допустимых решений Ω (рис. 6.3.1, c).

Как видно, решаемая задача однозначно определяется областью Ω и скалярной функцией качества $Q(U)$. Эта функция образует в Ω скалярное поле, которое удобно в случае $q = 2$ характеризовать линиями равного уровня минимизируемой функции. Уравнения этих линий

$$Q(U) = c = \text{const} \quad (6.3.2)$$

получаем при различных значениях параметра c . Легко заметить, что линии равного уровня на плоскости управляемых параметров $\{u_1, u_2\}$ представляют собой аналог

топографических линий равного уровня (например, над уровнем моря) на географической карте. Именно поэтому топографическая терминология используется при описании задач оптимизации. Такими топографическими терминами, широко применяемыми в теории и практике оптимизации, являются: «линии равного уровня» (6.3.2), которые образуют «рельеф» функции $Q(\mathbf{U})$, причем рельеф может быть разнообразным: «яма», «котловина», «впадина» (рис. 6.3.2,а), «холм» или «вершина»

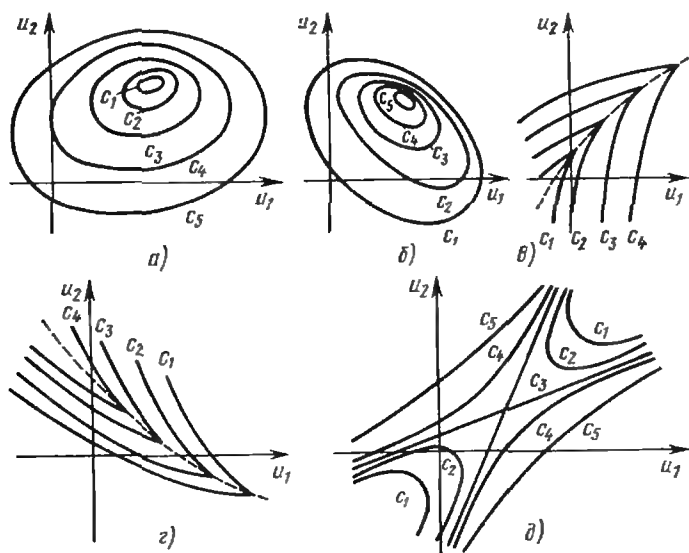


Рис. 6.3.2. Примеры «рельефов» функции качества ($c_1 < c_2 < c_3 \dots$)

(рис. 6.3.2,б), «овраг» или «щель» (рис. 6.3.2,в), «хребет» (рис. 6.3.2,г), «перевал» или «седло» (рис. 6.3.2,д), «плоскогорье» и т. д. Эта аналогия распространяется и на процесс решения оптимизационной задачи. Так, движение в направлении уменьшения функции качества называют «спуском», в направлении увеличения — «подъемом».

Естественно, что эту наглядную аналогию можно использовать и для многомерных задач оптимизации при $q > 2$. В этом случае при $q = 3$ линии равного уровня становятся поверхностями, а при $q > 3$ — гиперповерхностями равного уровня. Теперь рекуррентный процесс ре-

шения задачи математического программирования можно представить как движение в q -мерном пространстве $\{U\}$ от точки к точке в последовательности (6.3.1) (рис. 6.3.3).

Решение задач математического программирования можно осуществить по-разному. Можно использовать априорные сведения о структуре функций Q , H и C , т. е. построить алгоритмы, учитывающие специфику этих

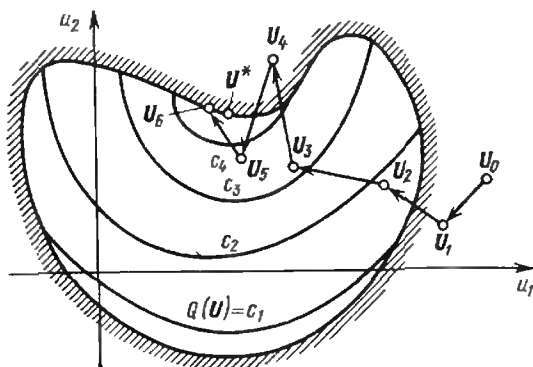


Рис. 6.3.3. Пример траектории решения задачи математического программирования ($c_1 > c_2 > c_3 \dots$)

функций, специальными методами (некоторые из них будут рассмотрены в следующем параграфе). Но для сложных объектов управления эту специфику выяснить и учесть очень трудно и проще поступить иначе — организовать процесс поиска решения U^* , используя лишь значения функций, а не сведения об их структуре. Такой подход называют поисковым методом оптимизации. Он имеет универсальный характер. Некоторые наиболее распространенные методы поисковой оптимизации описаны в приложении. Здесь же рассмотрим, каким образом априорные сведения о задаче могут быть «заложены» в алгоритме ее решения.

6.4. Некоторые специальные методы решения задач математического программирования

Рассмотрим некоторые методы, используемые при решении задач математического программирования. Особенности этих методов обусловлены прежде всего спецификой структуры функций Q , G и H .

Симплексный метод. Этот метод решения задач линейного программирования, для которых функции Q , G , H линейны по U :

$$Q(U) = \sum_{i=1}^n a_i u_i \rightarrow \min, \quad U \in \Omega \quad (6.4.1)$$

где

$$\Omega: \sum_{k=1}^m b_{jk} u_k \geq c_j, \quad j=1, \dots, k_2.$$

Эта запись отличается от (6.2.1), где еще есть ограничения типа равенств. Однако ввиду линейности их легко исключить, уменьшив тем самым число независимых переменных. С другой стороны, от ограничений типа равенств легко избавиться формальным образом, заменив $q_i(U)=0$ двумя: $q_i(U) \geq 0$ и $-q_i(U) \geq 0$. Будем считать, что одна из этих операций проделана и задача линейного программирования приобрела вид (6.4.1).

Число ограничений, образующих область допустимых решений Ω в задачах линейного программирования, всегда должно превышать число переменных, т. е.

$$k_2 > q, \quad (6.4.2)$$

иначе задача, вообще говоря, может не иметь решения. Заметим, что обычно q из них имеют вид

$$u_i \geq 0, \quad i=1, \dots, q, \quad (6.4.3)$$

т. е. решение ищется в положительном квадранте (это связано с физическим смыслом переменных: u_i — обычно количество). Введем обозначение

$$h_j = h_j(U) = \sum_{k=1}^m b_{jk} u_k - c_j, \quad j=1, \dots, k_2. \quad (6.4.4)$$

Тогда уравнение области Ω можно записать в стандартном виде

$$\Omega: h_j = h_j(U) \geq 0, \quad j=1, \dots, k_2. \quad (6.4.5)$$

Легко заметить, что область представляет собой выпуклый многогранник. Действительно, каждое из условий $h_j(U) \geq 0$ определяет полупространство. Пересечение всех этих полупространств может быть только или выпуклым, или пустым (при противоречивых условиях)

многогранником. На рис. 6.4.1 показан пример образования такого выпуклого многогранника для $q=2$.

Решение задачи линейного программирования (6.4.1) в силу линейности $Q(U)$ лежит на поверхности этого многогранника. Действительно, оно не может находиться внутри его, так как в этом случае можно было бы снять ограничения Ω . Но тогда $U^* \rightarrow \infty$ в силу линейности $Q(U)$, поэтому

$$U^* \in \bar{\Omega}, \quad (6.4.6)$$

где $\bar{\Omega}$ — граница многогранника Ω . Далее легко видеть, что решение всегда лежит на одной из вершин этого

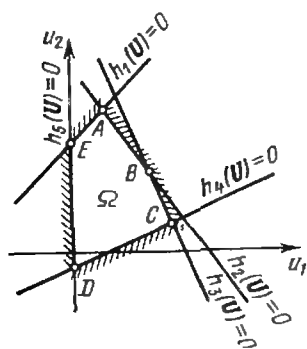


Рис. 6.4.1. Пример образования выпуклого многогранника Ω в задаче линейного программирования

многогранника или на грани. В последнем случае любая точка этой грани, в том числе и ее вершины, в силу линейности функции $Q(U)$ может считаться решением задачи (6.4.1). Поэтому для решения задачи достаточно рассмотреть только вершины многогранника Ω .

И последнее, так как каждая вершина многогранника Ω определяется в общем случае q из k_2 ограничений (6.4.5), то в каждой вершине q ограничений из (6.4.5) будут равны нулю, а остальные — положительны (точнее, неотрицательны).

Это позволяет все вершины кодировать вектором

$$H = (h_1, \dots, h_{k_2}), \quad (6.4.7)$$

составляющие которого равны значениям ограничений (6.4.5) в этой вершине. При этом вектор (6.4.7) имеет по крайней мере q нулевых компонент, а остальные составляющие положительны.

Например, для случая, показанного на рис. 6.4.1, получаем следующие коды вершин $A, \dots, E$:

$$H_A = (0, 0, h_3, h_4, h_5),$$

$$H_B = (h_1, 0, 0, h_4, h_5),$$

$$\dots$$

$$H_E = (0, h_2, h_3, h_4, 0).$$

Симплексный метод решения задачи линейного программирования заключается в переходе по граням многогранника Ω в соседние (смежные) вершины, где значение функции качества Q меньше, чем в исходной вершине. Алгоритм симплексного метода сводится к следующим операциям на каждом шаге.

1. Определить все вершины $U_N^{(i)}$ ($i=1, \dots, k_N$), смежные с исходной на N -м шаге, вершиной U_N (здесь k_N — число вершин, смежных с U_N).

2. Выбрать из них ту, в которой значение функции качества минимально:

$$Q(U_N^{(j)}) = \min_{i=1, \dots, k_N} Q(U_N^{(i)}).$$

3. Перейти в эту вершину, т. е.

$$U_{N+1} = U_N^{(j)},$$

и затем перейти к следующему $(N+1)$ -му шагу, т. е. искать вершины, смежные с U_{N+1} , и т. д. (см. п. 1). Работа алгоритма заканчивается, если среди всех вершин, смежных с U_N , не найдется ни одной с меньшим значением функции качества, чем в исходной. В этом случае $U^* = U_N$. Основная трудность симплексного метода заключается в определении смежных вершин. Эту трудность позволяют преодолеть так называемые симплекс-таблицы, о которых можно прочесть в любом руководстве по линейному программированию (см. комментарий к этой главе).

Метод возможных направлений. Этот метод используется для решения задач выпуклого программирования вида

$$Q(U) \rightarrow \min_{U \in \Omega}, \quad (6.4.8)$$

где

$$\Omega : h_j(U) \geq 0, \quad j=1, \dots, k_2;$$

$Q(U)$ и $h_j(U)$ — выпуклые вниз функции, обладающие свойством (6.2.5). Область Ω при этом выпукла (рис. 6.4.2), так как пересечение выпуклых областей образует выпуклую область.

Решение задачи (6.4.8), вообще говоря, может находиться как внутри области Ω , так и на ее границе.

Введем понятие возможного направления. Пусть U_N — некоторая точка, принадлежащая Ω ($U_N \in \Omega$). Тогда возможное направление определяется таким вектором Y , вдоль которого функция качества $Q(U)$ уменьшается без нарушения ограничений Ω , т. е.

$$Q(U_N + \alpha Y) < Q(U_N), \quad (6.4.9)$$

где $U_N + \alpha Y \in \Omega$ и $\alpha > 0$ мало.

Решение задачи определения возможного направления Y и составляет основную трудность реализации это-

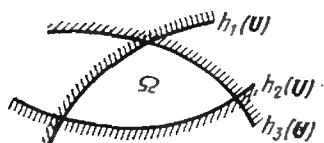


Рис. 6.4.2. Пример области, образованной выпуклыми вниз ограничениями

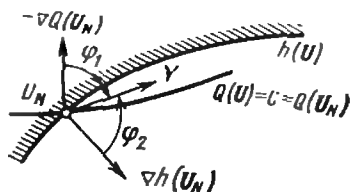


Рис. 6.4.3. Пример области, для которой антиградиентное направление не является возможным

го метода. Если исходная точка U_N находится внутри области Ω (т. е. не на ее границе), лучшим является антиградиентное направление

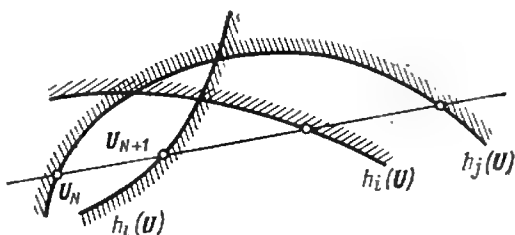
$$Y = -\nabla Q(U_N).$$

Если же U_N лежит на границе области Ω , антиградиентное направление $-\nabla Q(U_N)$ может оказаться невозможным (рис. 6.4.3, где ∇h — вектор градиента ограничений $h(U) \geq 0$ в точке U_N). Как хорошо видно, для возможного направления Y необходимо, чтобы углы φ_1 и φ_2 , образуемые им с векторами $-\nabla Q(U_N)$ и $\nabla h(U_N)$ были бы меньше $\pi/2$, т. е. $\varphi_i < \pi/2$ ($i=1, 2$). Этому условию всегда можно удовлетворить, если направления векторов $-\nabla Q$ и ∇h не противоположны.

Можно потребовать и большего. Так как возможных направлений Y существует целый набор, то естественно выбрать из них одно, в каком-то смысле оптимальное. Таким направлением Y^* будет то, при движении по ко-

торому функция качества $Q(U)$ уменьшается наиболее интенсивно, т. е. угол φ_1 минимален, и одновременно происходит наиболее быстрое удаление от ограничения $h(U)=0$, т. е. угол φ_2 тоже минимален. Эти противоречивые требования (так как сумма $\varphi_1 + \varphi_2$, как видно из рис. 6.4.3, постоянна) можно удовлетворить, решив соответствующую задачу линейного программирования (см. руководство по выпуклому программированию в комментариях к этой главе).

Рис. 6.4.4. К определению величины коэффициента α'_l



После определения оптимального возможного направления Y^* следует организовать спуск вдоль этого направления, т. е. определить такое значение α^* в $U = U_N + \alpha Y^*$, более которого α увеличить нельзя, так как при этом либо начинает увеличиваться функция качества $Q(U)$, либо нарушается одно из ограничений $h_j(U) \geq 0$ ($j=1, \dots, k_2$). Для этого следует решить задачи

$$Q(U_N + \alpha Y^*) \rightarrow \min_{\alpha > 0} \Rightarrow \alpha'_0 > 0, \quad (6.4.10)$$

$$h_l(U_N + \alpha Y^*) = 0 \Rightarrow \alpha'_l > 0, \quad j = 1, \dots, k_2,$$

и выбрать из всех α'_j ($j=0, \dots, k_2$) наименьшее:

$$\alpha^* = \min_{j=0, \dots, k_2} \alpha'_j = \alpha'_l. \quad (6.4.11)$$

Необходимость такого выбора хорошо видна из примера, приведенного на рис. 6.4.4, где $\alpha'_0 > \alpha'_l$.

На этом заканчивается N -й этап решения задачи выпуклого программирования $U_{N+1} = U_N + \alpha^* Y^*$, и следует обратиться к следующему $(N+1)$ -му, на котором для определения возможного направления необходимо вычислять градиент l -го ограничения (6.4.11).

Методы решения задач целочисленного программирования. Рассмотрим методы решения задач целочисленного программирования

$$Q(U) \rightarrow \min_{U \in \Omega_d} \Rightarrow U^*_d, \quad (6.4.12)$$

$$h_j(U) \geq 0, \quad j = 1, \dots, k_2,$$

$$u_i \in \{u_i^{(1)}, \dots, u_i^{(l_i)}\}, \quad i = 1, \dots, q,$$

где l_i — конечное число дискретных значений, которые может принимать i -й компонент вектора управления $U = (u_1, \dots, u_q)$. Как видно, множество Ω допустимых значений управления U конечно и имеет $\prod_{i=1}^q l_i$ элементов,

что при больших l_i или q исключает возможность полного перебора, т. е. решения задачи (6.4.12) «в лоб». Поэтому рассмотрим иные методы. Пусть для определенности $u_i \in \{1, \dots, l_i\}$.

Простейшим методом является решение непрерывной задачи

$$Q(U) \rightarrow \min_{U \in \Omega} \Rightarrow U^*,$$

$$\Omega : h_j(U) \geq 0, \quad i = 1, \dots, k_2,$$

которая получается из (6.4.12) путем снятия ограничения на целочисленность компонентов вектора U , и выбор в ближайшей окрестности полученного решения U^* целочисленного вектора U'_d , удовлетворяющего ограничениям Ω при минимальном значении функции качества Q . Для этого следует сначала найти ближайшую целочисленную точку, округлив до целого ее координаты (например, $2,1 \rightarrow 2$; $3,8 \rightarrow 4$), и проверить выполнение ограничений. Если ограничения выполняются, то эту точку и называют решением задачи целочисленного программирования, т. е. $U'_d = U^*_d$ (пример такой точки показан на рис. 6.4.5,а). Если же ближайшая целочисленная точка U'_d окажется вне допустимой области Ω , т. е. $U'_d \notin \Omega$ (этот случай показан на рис. 6.4.5,б), то следует рассмотреть ее целочисленную окрестность (она очерчена штрихами на рис. 6.4.5,б), содержащую $3^q - 1$ точек. Из этих точек прежде всего следует выделить под-

множество тех, которые удовлетворяют ограничениям Ω . Пусть это будут точки $U_1, \dots, U_k$. Тогда выбор одной из них U^*_d производится по минимальному значению функции качества, т. е.

$$Q(U^*_d) = \min_{i=1, \dots, k} Q(U_i).$$

Однако этот способ далеко не всегда приводит к точному решению. Такой случай показан на рис. 6.4.5, в, где решение целочисленной задачи лежит в точке A , а не

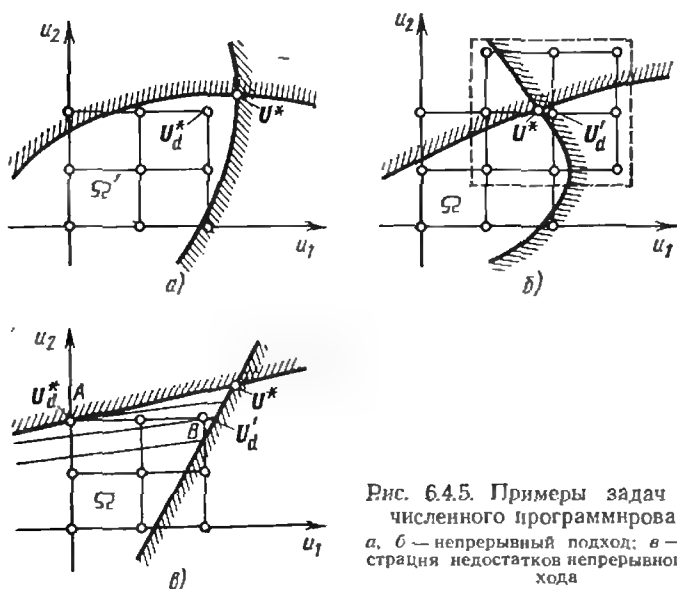


Рис. 6.4.5. Примеры задач целочисленного программирования:

a, b — непрерывный подход; v — иллюстрация недостатков непрерывного подхода

в B , полученной изложенным методом (здесь параллельными линиями обозначены линии равного уровня). Это обстоятельство заставляет воспользоваться другими методами решения задач целочисленного программирования. Рассмотрим два из них.

Прямой поиск опирается на идею дискретного случайного поиска, в процессе которого движение происходит по целочисленным точкам допустимой области Ω . Пусть U_N — некоторая целочисленная точка в Ω . Процесс случайного дискретного поиска заключается в том,

что случайно определяется точка целочисленного окружения U_N

$$U_N^{\xi} = U_N + \Xi,$$

где $\Xi = (\xi_1, \dots, \xi_q)$ — случайный целочисленный вектор с компонентами, принимающими независимо и равновероятно одно из трех значений $\xi_i \in \{+1; 0; -1\}$. Если значение функции качества Q в точке U_N^{ξ} оказалось меньше, чем в исходной точке U_N , то $U_{N+1} = U_N^{\xi}$. В противном случае (при неудачном шаге) определяется другой случайный вектор Ξ и снова проверяются условия уменьшения функции качества. Таким образом, алгоритм дискретного случайного поиска можно записать в виде

$$U_{N+1} = \begin{cases} U_N + \Xi, & \text{если } Q(U_N + \Xi) < Q(U_N), \\ U_N, & \text{если } Q(U_N + \Xi) \geq Q(U_N). \end{cases} \quad (6.4.13)$$

Как видно, этим осуществляется случайный перебор целочисленного окружения исходной точки до тех пор, пока не будет найдена точка с меньшим значением функции качества, чем у исходной. Заметим, что ввиду конечности множества случайных векторов $\{\Xi\}$ при неудачном шаге целесообразно исключать из дальнейшего рассмотрения на этом этапе те вектора Ξ , которые уже были. При этом за счет исключения ненужных повторений повышается эффективность поиска.

Для задач дискретного выпуклого программирования, когда $Q(U)$ и $H(U)$ — выпуклые функции, этот метод сходится, т. е. неизбежно приводит к решению поставленной задачи U^*_d .

Обратный поиск связан с движением из точки U'_d , ближайшей к U^* из допустимых, т. е. принадлежащих области Ω . Рассмотрим дискретное окружение этой точки, состоящее из $3^q - 1$ точек. Определяя все допустимые (т. е. принадлежащие Ω) точки, следует из них выбрать ту, в которой значение функции качества Q минимально. Если это наименьшее значение окажется больше, чем в исходной точке U'_d , то естественно считать U'_d решением задачи. Если же меньше, то следует рассмотреть целочисленную окрестность этой новой точки.

Обратный поиск решает задачи дискретного выпуклого программирования так же успешно, как и прямой.

Однако для его осуществления требуется предварительно решить непрерывную задачу. Обратный поиск от прямого отличается полным перебором окрестности каждой точки, претендующей на оптимум. Но таких полных переборов при обратном поиске будет значительно меньше, чем случайных переборов при прямом поиске.

В сложных случаях можно рекомендовать оба метода и выбрать то решение, которое имеет наименьшее значение функции качества.

6.5. Некоторые специальные методы решения вариационных задач

В § 6.1 показано, как задачу синтеза управления динамическим объектом можно свести к задаче математического программирования (6.1.8). Для этого достаточно представить искомое управление $U(t)$ в виде набора дискрет:

$$U_i = U(t_i) \quad i=1, \dots, k,$$

определение которых можно осуществить, решив указанную задачу математического программирования относительно векторных параметров $U_1, \dots, U_k$ (всего при этом приходится определять kq значений параметров, образующих оптимальное управление).

Однако при этом подходе получается, как правило, очень сложная многоэкстремальная задача математического программирования, решение которой обычно вызывает большие трудности. Именно поэтому для решения вариационных задач часто удобнее применять специальные методы, два из которых мы рассмотрим ниже.

Принцип максимума. Этот метод применяется для динамических объектов, модель которых может быть записана в виде системы обыкновенных дифференциальных уравнений

$$\dot{y}_i = f_i(y_1, \dots, y_m, u(t)), \quad i=1, \dots, m, \quad (6.5.1)$$

с заданными начальными условиями при $t=0$ $y_i(0) = y_i^0$ ($i=1, \dots, m$) и скалярным управлением. Удобно систему (6.5.1) записать в векторной форме:

$$\dot{Y} = F(Y, u(t)),$$

где $Y(t) = (y_1(t), \dots, y_m(t))$ — вектор переменных задачи, а $F = (f_1, \dots, f_m)$ — вектор заданных функций, которые определяют модель объекта.

Необходимо решить следующую задачу: перевести объект из состояния $Y_0 = (y_1(0), \dots, y_m(0))$ в заданное состояние

$$Y^* = (y^*_1, \dots, y^*_m), \quad (6.5.2)$$

причем траектория $Y(t)$ и управление $u(t)$ должны быть такими, чтобы удовлетворить заданным ограничениям и экстремальной цели — минимуму заданного функционала

$$Q(u(t)) = \int_0^T f_0(Y(t), u(t)) dt, \quad (6.5.3)$$

где $f_0(Y(t), u(t))$ — заданная функция. В качестве такого функционала обычно выбирают всякого рода затраты на процесс управления. При $f_0(\cdot) = 1$ получаем $Q = T$ — время перехода объекта из состояния Y_0 в Y^* .

Примером типичной задачи такого рода является задача крановщика. Нужно переместить тяжелый груз из одной точки в другую за минимальное время. Крановщик располагает трюичным управлением $u = \{-1; 0; 1\}$, т. е. кран может двигаться в одну ($u = -1$) или другую ($u = 1$) сторону и остановиться ($u = 0$). Неопытный крановщик тем и отличается, что обычно раскачивает груз и долго не может его «успокоить». Оптимальное управление $u^*(t)$ позволяет ему за минимальное время T остановить груз точно в заданной точке Y^* .

Таким образом, поставленная вариационная задача формулируется в виде

$$\int_0^T f_0(Y(t), u(t)) dt \rightarrow \min_{u(t) \in \Omega} \quad (6.5.4)$$

$$\Omega: \begin{cases} h_j(Y(t), u(t)) \geq 0, & j = 1, \dots, k_2, \\ \dot{y}_i = f_i(Y(t), u(t)), & i = 1, \dots, m, \\ y_i(0) = y^0_i, \end{cases}$$

где $h_j(\cdot)$ — заданные функции ограничений типа неравенств. Заметим, что в формулировке (6.5.4) не оговорены граничные условия (6.5.2): они появятся позже. А пока сформулирована задача о переводе объекта в какое-то, (пока!) не интересующее нас состояние $Y(T)$.

Принцип максимума позволяет задачу (6.5.4) свести к задаче максимизации так называемой функции Гамильтона:

$$\Gamma(u(t)) = \sum_{i=0}^m \psi_i(t) f_i(Y(t), u(t)) \rightarrow \max_{u(t) \in \Omega'} \quad (6.5.5)$$

при ограничениях

$$\Omega' : h_j(Y(t), u(t)) \geq 0, \quad i=1, \dots, k_2, \quad (6.5.6)$$

где функции $\psi_i(t)$ определяют из решения системы $2m+1$ дифференциальных уравнений вида

$$\begin{aligned} \dot{\psi}_0 &= 0, \quad \dot{\psi}_i = - \sum_{j=0}^m \psi_j \frac{\partial f_j(Y(t), u(t))}{\partial y_i}, \\ \dot{y}_i &= f_i(Y(t), u(t)), \quad i=1, \dots, m \end{aligned} \quad (6.5.7)$$

с начальными условиями

$$\begin{aligned} y_j(0) &= y_j^0, \quad j=1, \dots, m, \\ \psi_i(0) &= c_i, \quad i=0, \dots, m. \end{aligned} \quad (6.5.8)$$

Эта задача интересна тем, что максимизация может производиться в каждый момент времени t , так как $\Gamma(u(t))$ не функционал, а функция и задача максимизации одномерная по параметру управления u . При этом минимизируемая функция изменяется определенным образом во времени [см. (6.5.7)] при ограничениях (6.5.6), причем минимизацию следует осуществлять в каждый момент времени t .

Решение этой задачи при заданных начальных условиях (6.5.8) не представляет труда. Действительно, решение сводится к интегрированию системы дифференциальных уравнений (6.5.7) и решению задачи одномерной оптимизации (6.5.5) на каждом шаге интегрирования (например, методом золотого сечения, описанным в приложении).

Теперь рассмотрим вопрос о попадании в заданную точку Y^* . Очевидно, что состояние $Y(t)$ объекта в момент t определяется начальными условиями $C = (c_0, \dots, c_m)$, заданными (6.5.8), т. е. в результате решения задачи (6.5.5) получаем $Y(T)$ в зависимости от значений $C : Y(T) = f(C)$. Теперь задача определения начальных условий C формулируется просто: их нужно подобрать

так, чтобы $f(C) = Y^*$. Но зависимость $f(C)$ не явна, она получается алгоритмически в результате решения задачи (6.5.5). Поэтому процедуру определения C естественно свести к решению минимизационной задачи:

$$Q(C) = |f(C) - Y^*| \rightarrow \min_C \Rightarrow C^*,$$

т. е. нужно подобрать такие начальные условия C , чтобы функция невязки $Q(C)$ была минимальной (в идеале равной нулю). Эту задачу следует решать поисковыми методами, рассмотренными в приложении. При

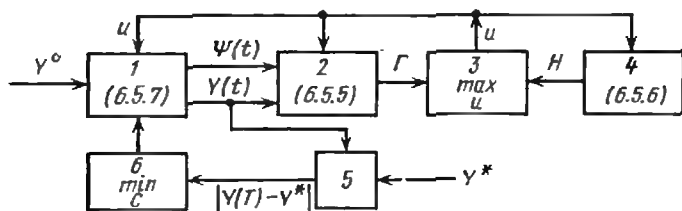


Рис. 6.5.1. Схема вычислений по принципу максимума

этом определение каждой невязки Q в новой точке C связано с решением задачи (6.5.5). При определении C^* , такого, что $Q(C^*) = 0$, получаем решение задачи (6.5.5) в виде оптимального управления $u^*(t)$.

Как легко заметить, эту процедуру легко распространить на случай многомерного управления $U(t) = (u_1(t), \dots, u_q(t))$. При этом задача минимизации (6.5.5) становится многопараметрической, и ее следует решать одним из поисковых методов (см. приложение).

Рассмотрим схему реализации вычислений по принципу максимума. Она показана на рис. 6.5.1. Здесь 1 — блок решения системы дифференциальных уравнений (6.5.7), на вход которого подаются управление $u(t)$ и начальные условия C и $Y^0 = (y^0_1, \dots, y^0_m)$. Решение этой системы уравнений $Y(t)$ и $\Psi(t) = (\psi_0(t), \dots, \psi_m(t))$ подается на блок 2, где вычисляется значение функции Гамильтона по формуле (6.5.5), которая все время максимизируется блоком 3 по u с учетом ограничений (6.5.6), где $H = (h_1, \dots, h_{k_2})$ (это контур быстрой оптимизации). Имеется контур медленной оптимизации, который решает задачу минимизации функции невязки

$|Y(T) - Y^*|$ путем изменения начальных условий C . Для этого на каждом этапе вычисляется невязка $Y(T)$ с целевым состоянием Y^* (блок 5). Один шаг минимизации (блок 6) здесь совершается после однократного решения системы (6.5.7). Реализация такой системы на современных ЭВМ в принципе не представляет труда.

Однако функция качества $Q(C)$ задачи внешней (медленной) оптимизации обычно имеет многоэкстремальный характер, что заставляет обращаться к глобальным методам поиска (изложенным в приложении), которые обычно требуют при своей реализации значительных затрат машинного времени. Причем эти затраты сильно зависят от размерности m объекта. Это обстоятельство не позволяет использовать метод для решения задач синтеза управления сложными объектами, имеющими большую размерность m .

Динамическое программирование. Метод динамического программирования применяют обычно для многоэкстремальных вариационных задач, решение которых в той или иной степени сводится к организации полного перебора. Этот метод позволяет упорядочить и тем самым значительно сократить полный перебор всех допустимых вариантов управления. Это оказывается возможным потому, что предполагается выполненным так называемый «принцип оптимальности». Смысл его сводится к тому, что оптимальное поведение зависит только от исходного состояния объекта и не зависит от того, какова была предыстория попадания в это исходное состояние. Так, управляя каким-то процессом, следует интересоваться его состоянием только в данный момент, так как именно от этого зависит дальнейшее управление, а не от его предыстории.

Естественно, что этот принцип применим для достаточно простых динамических систем, которые мы и рассматриваем здесь. «Простота» здесь в том, что уже имеется математическая модель объекта управления, позволяющая достаточно хорошо определять будущее поведение объекта при том или ином управлении.

Поясним суть метода динамического программирования на примере решения следующей вариационной задачи, которая является частным случаем (6.5.4):

$$Q(u(t)) = \int_0^t f_0(y(\bar{t}), u(\bar{t})) d\bar{t} \rightarrow \min_{u(t) \in \mathfrak{R}}, \quad (6.5.9)$$

где

$$\Omega: \begin{cases} \omega_1(t) \leq u(t) \leq \omega_2(t), \\ \frac{dy}{dt} = f_1(y, u(t)), \quad y(0) = y_0; \\ y(T) = y^*, \end{cases} \quad (6.5.10)$$

где функции ограничения $\omega_1(t)$ и $\omega_2(t)$ заданы.

Задача (6.5.9), таким образом, состоит в определении оптимального управления $u^*(t)$ ($0 \leq t \leq T$) объектом первого порядка [см. (6.5.10)], поведение которого $y(t)$ минимизирует заданный функционал качества Q (6.5.9).

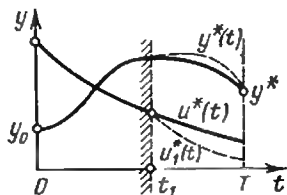


Рис. 6.5.2. К иллюстрации принципа оптимальности

На рис. 6.5.2 показано решение этой задачи — оптимальное управление $u^*(t)$ и оптимальное поведение объекта $y^*(t)$ на всем интервале. Теперь рассмотрим какой-то момент t_1 и применим принцип оптимальности, согласно которому оптимальное управление

с момента t_1 будет зависеть только от состояния в этот момент, т. е. $y^*(t_1)$. Это очевидно. Естественно задать вопрос: а совпадут ли оптимальные управления, построенные из $t=0$ и $t=t_1$ на интервале $t_1 \leq t \leq T$, не изменится ли при этом оптимальное управление, а вслед за ним и поведение объекта так, как, например, показано штрихом на рис. 6.5.2? Ответ на этот не столь тривиальный вопрос следует из принципа оптимальности: нет, не изменится!

Действительно, рассмотрим две реализации одного и того же способа решения исходной задачи, т. е. определения оптимального управления на интервале $[0, T]$. Пусть в первом случае мы получили оптимальное решение $u^*(t)$ ($0 \leq t \leq T$). Теперь разобьем решение задачи на два этапа: $[0, t_1]$ и $[t_1, T]$ и снова решим задачу тем же способом. Пусть полученное при этом оптимальное управление равно $u^*_1(t)$ ($t_1 \leq t \leq T$), отлично от $u^*(t)$ (см. штриховую линию на рис. 6.5.2). Но результат решения задачи не должен зависеть от способа ее решения (если, разумеется, способы правильные). Поэтому

$$u^*_1(t) = u^*(t), \quad t_1 \leq t \leq T, \quad (6.5.11)$$

т. е. любая часть $[t_1, T]$ оптимальной траектории оптимальна. Это не очень очевидное следствие принципа

оптимальности и лежит в основе вычислительной процедуры динамического программирования. Заметим, что из (6.5.11) вовсе не следует, что любой отрезок $[t_1, t_2]$ оптимальной траектории оптимален с точки зрения минимума функционала качества на этом промежутке. Если $t_2 \neq T$, то это не так. Действительно, если бы оптимальность на отрезке $[t_1, t_2 < T]$ соответствовала оптимальности на всем интервале $[0, T]$, то это означало бы, что управление на $[t_1, t_2]$ не влияет на будущее поведение объекта. Это бывает лишь в очень редких случаях, которые, как правило, неинтересны, так как задача при этом вырождается.

Теперь обратимся к методу динамического программирования. Прежде всего сделаем задачу (6.5.9) дискретной (это упростит алгоритм вычислений), т. е. будем рассматривать управление лишь в дискретные моменты времени:

$$Q(U) = \sum_{i=0}^{N-1} f_0(y_i, u_i) \Delta t_i \rightarrow \min_{u_i \in \Omega}, \quad (6.5.12)$$

$$\Omega: \begin{cases} \omega'_i \leq u_i \leq \omega''_i, & i=0, \dots, N, \\ y_{i+1} = y_i + f_1(y_i, u_i) \Delta t_i, \\ y_0 = y^0, y_N = y^*, & i=0, \dots, N-1, \end{cases}$$

причем $u_i = u(t_i)$; $y_i = y(t_i)$; $\Delta t_i = t_{i+1} - t_i$; $\omega'_i = \omega_1(t_i)$; $\omega''_i = \omega_2(t_i)$.

Для простоты рассмотрим случай равноотстоящих единичных моментов времени, т. е. $\Delta t = 1$ ($i=0, \dots, N-1$). Получаем

$$Q(U) = \sum_{i=0}^{N-1} f_0(y_i, u_i) \rightarrow \min_{u_i \in \Omega}, \quad (6.5.13)$$

$$\Omega: \begin{cases} \omega'_i \leq u_i \leq \omega''_i, \\ y_{i+1} = y_i + f_1(y_i, u_i), \\ y_0 = y^0; y_N = y^*. \end{cases}$$

Для решения этой задачи воспользуемся принципом оптимальности, записанным в виде (6.5.11). Будем искать решение «с конца» ($t=T$). Принцип оптимальности гарантирует нам, что оптимальное решение, полученное на отрезке $[t_1, T]$, будет «куском» решения исходной задачи на $[0, T]$.

Пусть условия Ω допускают управление на последнем $(N-1)$ -м шаге в пределах Ω_{N-1} : $\omega'_{N-1} \leq u_{N-1} \leq \omega''_{N-1}$ (рис. 6.5.3). На этом последнем шаге решается задача попадания в точку $y_N = y^*$, т. е. задача определения корня уравнения

$$y^* = y_{N-1} + f_1(y_{N-1}, u_{N-1}) \Rightarrow u^*_{N-1}(y_{N-1}), \quad (6.5.14)$$

решением которой является значение управления u^*_{N-1} в зависимости от y_{N-1} и независимо от того, какое значение примет при этом мини-

минизируемый функционал Q . Действительно, на этом последнем шаге не до оптимизации: нужно удовлетворить граничному условию $y_N = y^*$ любой ценой. Эта «цена» зависит от y_{N-1} и равна

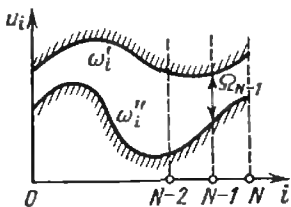


Рис. 6.5.3. Область допустимых управлений

$$\varphi_{N-1}(y_{N-1}) = f_0(y_{N-1}, u^*_{N-1}) \quad (6.5.15)$$

(напомним, что f_0 — функция затрат). Запомним это обстоятельство.

Теперь, двигаясь от конца ($t=T$) к началу ($t=0$), рассмотрим следующий $(N-2)$ -й шаг (предпоследний), который делается из пока неизвестного состояния y_{N-2} . Прирост минимизируемого критерия на этом шаге равен $f_0(y_{N-2}, u_{N-2})$. В соответствии с принципом оптимальности следует минимизировать не эту величину, а все затраты до конца процесса, т. е. сумму затрат на этом и последнем шаге. Но для последнего шага затраты известны [см. (6.5.15)]. Следовательно, надо минимизировать сумму

$$f_0(y_{N-2}, u_{N-2}) + \varphi_{N-1}(y_{N-1}) \rightarrow \min_{u_{N-2} \in \Omega_{N-2}}, \quad (6.5.16)$$

где

$$y_{N-1} = y_{N-2} + f_1(y_{N-2}, u_{N-2}). \quad (6.5.17)$$

Однако начальные условия этого этапа, т. е. величина y_{N-2} , нам как раз и неизвестны. Поэтому следует «заготовить впрок» оптимальное управление u^*_{N-2} для этого шага в зависимости от неизвестного значения y_{N-2} . Для этого следует решить задачу (6.5.16) с учетом (6.5.17).

Обозначим минимальное значение затрат на последних двух этапах функцией

$$\varphi_{N-2}(y_{N-2}) = \min_{u_{N-2} \in \Omega_{N-2}} \{f_0(y_{N-2}, u_{N-2}) + \varphi_{N-1}(y_{N-2} + f_1(y_{N-2}, u_{N-2}))\}. \quad (6.5.18)$$

Поступая аналогично на k -м от конца шаге, получаем выражение суммарных затрат для последних k этапов

$$f_0(y_{N-k}, u_{N-k}) + \varphi_{N-(k-1)}(y_{N-(k-1)}).$$

Минимизируя это выражение по u_{N-k} с учетом того, что

$$y_{N-(k-1)} = y_{N-k} + f_1(y_{N-k}, u_{N-k}),$$

получаем $u_{N-k}^*(y_{N-k})$ и определяем в результате функцию минимальных затрат:

$$\varphi_{N-k}(y_{N-k}) = \min_{u_{N-k} \in \Omega_{N-k}} \{f_0(y_{N-k}, u_{N-k}) + \varphi_{N-(k-1)}(y_{N-k} + f_1(y_{N-k}, u_{N-k}))\}, \quad (6.5.19)$$

где Ω_{N-k} — множество допустимых управлений на k -м от конца шаге.

Таким образом, двигаясь «с конца», мы получаем следующие необходимые в дальнейшем функции:

$$\varphi_1(y_1), u_1^*(y_1), \dots, u_{N-1}^*(y_{N-1}), \quad (6.5.20)$$

которые характеризуют минимальные затраты (минимальное значение функционала качества Q) при движении из исходного состояния y^0 в целевое y^* и зависимость оптимального управления каждого шага от его начальных условий.

Теперь синтез оптимального управления очень прост — он идет «вперед» с начала. На первом шаге оптимальное управление определяется исходя из естественного условия

$$f_0(y_0, u_0) + \varphi_1(y_0 + f_1(y_0, u_0)) \rightarrow \min_{u_0 \in \Omega_0} \Rightarrow u_0^*,$$

$$y_1^* = y_0 + f_1(y_0, u_0^*)$$

— точка оптимальной траектории на первом шаге. Далее, используя полученные функции (6.5.20), имеем

$u^*_1 = u^*_1(y^*_1)$, $y^*_2 = y^*_1 + f_1(y^*_1, u^*_1)$ и т. д. до $u^*_{N-1} = u^*_{N-1}(y^*_{N-1})$.

Таким образом, определено оптимальное управление

$$U^* = (u^*_0, u^*_1, \dots, u^*_{N-1})$$

и соответствующая ему оптимальная траектория

$$Y^* = (y_0, y^*_1, \dots, y^*_{N-1}, y^*).$$

Проиллюстрируем применение метода на одном простом примере.

Пример. Найти оптимальное управление $u(t)$, ограниченное $|u(t)| \leq 1$ и переводящее объект $dy/dt = y + u$ из состояния $y(0) = 0$ в состояние $y(3) = 1$ ($T=3$) с минимальными затратами, вычисляемыми с помощью функционала

$$Q(u(t)) = \int_0^3 (y^2 + u^2) dt,$$

т. е. необходимо решить следующую вариационную задачу:

$$Q = \int_0^3 (y^2 + u^2) dt \rightarrow \min_{u(t) \in \Omega},$$

$$\Omega: \begin{cases} |u(t)| \leq 1, \\ \frac{dy}{dt} = y + u, \\ y(0) = 0; \quad y(3) = 1. \end{cases}$$

Запишем эту задачу в дискретной форме, предполагая, что $t_0=0$, $t_1=1$, $t_2=2$, $t_3=3$ и $y_i = y(t_i)$, $i=0, 1, 2, 3$, т. е. разбивая весь интеграл $[0, T]$ на три участка:

$$Q = \sum_{i=0}^2 (y_i^2 + u_i^2) \rightarrow \min_{U \in \Omega} \Rightarrow U^* = (u^*_0, u^*_1, u^*_2),$$

$$\Omega: \begin{cases} -1 \leq u_i \leq 1, \\ y_{i+1} = 2y_i + u_i \quad (i=0, 1, 2), \quad y_0=0, \quad y_3=1. \end{cases}$$

Это означает, что в данном случае $N=3$, $\omega'_i = -1$, $\omega''_i = 1$,

$$f_0(y_i, u_i) = y_i^2 + u_i^2, \quad f_1(y_i, u_i) = y_i + u_i, \\ y^0 = 0, \quad y^* = 1.$$

Начнем с конца ($t_2=2$). Определим управление, переводящее объект из состояния y_2 в $y^*=1$:

$$y^* = 1 = 2y_2 + u^*_2.$$

Далее определим оптимальное управление на последнем шаге (второй шаг от начала)

$$u^*_2(y_2) = 1 - 2y_2 \quad (*)$$

и минимальные затраты

$$\varphi_2(y_2) = y_2^2 + [u^*_2(y_2)]^2 = 5y_2^2 - 4y_2 + 1.$$

Затем образуем затраты на первом шаге

$$\varphi_1(y_1) = y_1^2 + u^*_1 + \varphi_2(y_2),$$

где $y_2 = 2y_1 + u_1$.

Получаем суммарные затраты (от $t=1$ до $t=3$)

$$\varphi_1(y_1) = y_1^2 + u^*_1 + 5(2y_1 + u_1)^2 - 4(2y_1 + u_1) + 1.$$

Минимизируя эту функцию по u_1 , определяем зависимость

$$u^*_1(y_1) = (1 - 5y_1)/3 \quad (**)$$

и минимальные затраты

$$\varphi_1(y_1) = (13y_1^2 - 4y_1 + 1)/3. \quad (***)$$

Теперь, располагая полученными зависимостями, можно строить оптимальное управление «вперед»:

1-й шаг. Подставляя в (***) выражение для $y_1 = 2y_0 + u_0 = u_0$ (так как $y_0 = 0$) и решая задачу

$$u_0^2 + \varphi_1(u_0) \rightarrow \min_{|u_0| \leq 1}$$

получаем $u^*_0 = 1/8$.

2-й шаг. Определяем первую точку оптимальной траектории

$$y^*_1 = 2y_0 + u^*_0 = u^*_0 = 1/8.$$

Подставляя это значение в (**), получаем оптимальное управление на 2-м шаге

$$u^*_1 = (1 - 5/8)/3 = 1/8.$$

и точку траектории

$$y^*_2 = 2y^*_1 + u^*_1 = 3/8.$$

3-й шаг. И, наконец, получаем из (*)

$$u^*_2 = 1 - 2y^*_2 = 1/4, \quad y^*_3 = 2y^*_2 + u^*_2 = 1 = y^*.$$

т. е. объект вышел в заданную точку.

Определяем, таким образом, оптимальное управление $U^* = (1/8, 1/8, 1/4)$ и оптимальную траекторию $Y^* = (0, 1/8, 3/8, 1)$. Затраты при этом равны

$$J^* = f_0(y_0, u^*_0) + \varphi_1(y^*_1) = 1/4.$$

Следует отметить, что ввиду простоты поставленной задачи минимизацию суммарных затрат на каждом шаге можно было сделать аналитически и получить зависимость оптимального управления от состояния в виде формулы (см. (**)) в рассмотренном примере). В более сложных случаях это сделать не удастся и приходится зависимость

$$u^*_i(y_i) \quad (6.5.21)$$

представлять таблично, что, естественно, снижает вычислительную эффективность метода.

Подведем итоги. Как было показано, метод динамического программирования является весьма эффективным средством решения не очень сложных вариационных задач. Он позволяет достаточно просто учитывать не только ограничения на управление $u_i \in \Omega_i$, но и ограничения, накладываемые на оптимальную траекторию $y^*(t)$, т. е. проводить решение даже через заданный «коридор». Для этого достаточно на каждом шаге решать задачу оптимизации не для всех значений координат, а лишь для попавших в коридор на этом шаге.

Таким образом, изложенная методика динамического программирования позволяет решать вариационные задачи вида

$$Q(U(t)) = \int_0^T f_0(Y(t), U(t)) dt \rightarrow \min_{U(t) \in \Omega}, \quad (6.5.22)$$

где

$$\Omega: \begin{cases} U(t) \in \omega(t), \\ Y(t) \in \xi(t), \\ \frac{dY}{dt} = F(Y, U), \\ Y(0) = Y^0; Y(T) = Y^*, \end{cases}$$

$$\text{а } Y = (y_1, \dots, y_n); \quad U = (u_1, \dots, u_m)$$

и заданы области $\omega(t)$ — допустимого управления $\xi(t)$ — допустимой траектории, которые могут определенным образом изменяться во времени, а также модель F и граничные условия Y_0 и Y^* задачи.

Одним из существенных преимуществ метода является гарантия того, что полученное решение представляет собой глобальный минимум функционала Q . Дело в том, что этот функционал обычно многоэкстремален при вариации управления $u(t)$. Поэтому решение поставленной задачи без применения принципа оптимальности заставляет обращаться и к поисковым методам глобальной оптимизации (см. приложение), которые всегда требуют значительных затрат машинного времени. Изложенный подход снимает проблему многоэкстремальности данной задачи.

Следует указать и на недостатки метода динамического программирования. К ним относится его практиче-

ская непригодность при решении задач управления объектами размерности более чем $m=2, 3$. В этом случае таблица (6.5.21) становится многомерной, что значительно снижает эффективность метода, так как сильно увеличивается объем необходимой памяти (примерно на порядок для каждой единицы размерности). Это обстоятельство исключает применение этого метода для управления объектами большой размерности, которыми обычно бывают сложные объекты управления.

В заключение этого параграфа отметим, что решение задачи синтеза оптимального управления динамическим объектом сводится к решению вариационной задачи, сложность которой зависит от размерности и сложности модели объекта. При этом решение полученной задачи управления сложным объектом целесообразно сводить к решению задачи математического программирования и в редких простых случаях к рассмотренным методам, опирающимся на идею динамического программирования и принципа максимума, описанных в этом параграфе. Следует помнить, что для указанных методов существует великолепно разработанная теория, которая, увы, не помогает решать сложных задач большой размерности, порождаемых практикой управления сложными системами. Поэтому при синтезе управления сложным динамическим объектом следует пользоваться методами решения многомерных и многоэкстремальных задач математического программирования, например специально для этого разработанными методами случайного поиска (см. приложение). Хорошая теория при решении такого рода задач является лишь компасом, а не инструментом и орудием. Решать же задачи приходится, используя аппарат поисковых методов оптимизации.

ЛИТЕРАТУРНЫЙ КОММЕНТАРИЙ К ГЛАВЕ 6

Задача синтеза управления при заданной цели и имеющейся модели объекта управления является задачей прикладной математики, так как сводится к решению задачи математического программирования (это показано в § 6.1), для решения которой разработаны необходимые вычислительные схемы.

Предлагаемая в § 6.2 классификация задач математического программирования, естественно, исполняя, как и всякая классификация. Ее цель ориентировать читателя относительно некоторой спецификации различных типов задач математического программирования, используемой при решении такого рода задач. Более подробную классификацию задач математического программирования можно найти в книгах [1—5].

Существует много методов решения математического программирования. Некоторые из них рассмотрены в § 6.4. Все они используют различную специфику функций Q , N и G . Число руководств по математическому программированию, пожалуй, даже слишком велико. Для первого ознакомления читателю целесообразно обратиться к [6—8].

Алгоритмы решения задач линейного программирования хорошо описаны в руководствах [1—6, 8—19, 22—24]. Задачи выпуклого программирования анализируются и решаются в книгах [9, 15, 20, 25]. С методами решения задач целочисленного программирования можно подробнее ознакомиться по соответствующим главам в книгах [1—3, 13, 21, 24—28]. Кроме того, существуют задачи геометрического программирования, с которыми можно познакомиться по [29, 30].

Существующие методы решения вариационных задач, рассмотренные в § 6.5, отличаются громоздкостью алгоритма и узостью класса решаемых им задач. Поэтому универсальным подходом к их решению является по-прежнему сведение к соответствующей задаче математического программирования [31]. Однако в отдельных частных случаях изложенные здесь методы могут дать значительный эффект. Более подробно с этими методами можно ознакомиться по руководствам [1—5, 7, 9, 12, 26, 32].

СПИСОК ЛИТЕРАТУРЫ

1. Кузин Л. Т. Основы кибернетики. — М.: Энергия, — Т. 1. Математические основы кибернетики, 1973. — Т. 1.
2. Раскин Л. Г. Анализ сложных систем и элементы теории управления. — М.: Сов. радио, 1976.
3. Зайченко Ю. П. Исследование операций. — Киев: Вища школа, 1975.
4. Основы кибернетики: математические основы кибернетики/ Под ред. К. А. Пупкова. — М.: Высшая школа, 1974.
5. Интрилигатор М. Математические методы оптимизации и экономическая теория: Пер с англ. — М.: Прогресс, 1975.
6. Гасс С. Путешествие в страну линейного программирования: Пер. с англ. — М.: Мир, 1973.
7. Ля Т. Г., Дамс Г. Э., Гейиз У. М. Управление процессами с помощью ЭВМ. Моделирование и оптимизация: Пер. с англ. — М.: Сов. радио, 1972.
8. Кофман А. Методы и модели исследования операций: Пер. с франц. — М.: Мир, 1966.
9. Габасов Р., Кириллова Ф. М. Методы оптимизации. — Минск: Белорусс. гос. ун-т, 1975.
10. Юдин Д. Б., Гольштейн Е. Г. Линейное программирование: Теория, методы и приложения. — М.: Наука, 1969.
11. Гасс С. Линейное программирование: методы и приложения: Пер. с англ. — М.: Физматгиз, 1961.
12. Коршунов Ю. М. Математические основы кибернетики. — М.: Энергия, 1972.
13. Линейное и нелинейное программирование/ Под ред. И. Н. Ляшенко. — Киев: Вища школа, 1975.
14. Вагнер Г. Основы исследования операций. — М.: Мир, 1972. Т. 1.

15. Зуховицкий С. И., Авдеева Л. И. Линейное и выпуклое программирование. — М.: Наука, 1967.
16. Ромакин М. И. Элементы линейной алгебры и линейного программирования. — М.: Высшая школа, 1963.
17. Карпелевич Ф. И., Садовский Л. Е. Элементы линейной алгебры и линейного программирования. — М.: Физматгиз, 1963.
18. Карлин С. Математические методы в теории игр, программировании и экономике: Пер. с англ. — М.: 1964.
19. Басов А. С. Линейное программирование в технико-экономических задачах. — М.: Наука, 1964.
20. Фиакко А., Мак-Кормик Г. Нелинейное программирование: Методы последовательной безусловной оптимизации: Пер. с англ. — М.: Мир, 1972.
21. Корбут А. А., Финкельштейн Ю. Ю. Дискретное программирование. — М.: Наука, 1969.
22. Юдин Д. Б., Гольштейн Е. Г. Линейное программирование — М.: Физматгиз, 1963.
23. Гейл Д. Теория линейных экономических моделей: Пер. с англ. — М.: ИЛ, 1963.
24. Мукачева Э. А., Рубинштейн Г. Ш. Математическое программирование. — Новосибирск: Наука, 1977.
25. Саати Т. Целочисленные методы оптимизации и связанные с ними экстремальные проблемы: Пер. с англ. — М.: Мир, 1973.
26. Вагнер Г. Основы исследования операций в 3-х томах. — М., Мир, 1973. — Т. 2.
27. Лихтенштейн В. Е. Дискретность и случайность в экономико-математических задачах. — М.: Наука, 1973.
28. Лихтенштейн В. Е. Модели дискретного программирования. — М.: Наука, 1971.
29. Зенер К. Геометрическое программирование и техническое проектирование: Пер. с англ. — М.: Мир, 1973.
30. Даффин Р., Питерсон Э., Зенер К. Геометрическое программирование: Пер. с англ. — М.: Мир, 1972.
31. Табак А., Куо Б. Оптимальное управление и математическое программирование. — М.: Наука, 1975.
32. Монсеев Н. Н. Численные методы в теории оптимальных систем. — М.: Наука, 1971.

Г Л А В А 7

РЕАЛИЗАЦИЯ УПРАВЛЕНИЯ

После решения задачи синтеза оптимального управления $U^*(t)$, которая обсуждалась в предыдущей главе, можно приступить к этапу управления. Это означает, что управляемые входы объекта U должны быть переведены в состояние, предписанное принятым решением $U^*(t)$. Для простых объектов управления в автоматическом варианте эта задача решается просто — путем «отработки» команды (программы) $U^*(t)$ с помощью соот-

ветствующих сервомеханизмов, автоматизирующих процесс изменения управляемых входов U объекта. В «ручном» варианте управляющее устройство выступает в роли советчика, указывающего, в какое положение следует перевести рукоятки управления объектов, а сама реализация осуществляется персоналом, обслуживающим этот объект. Здесь проблем не возникает, если, разумеется, нет сведений о неправильной работе управляющего устройства. Проблемы появляются лишь при управлении сложным объектом. Рассмотрим их.

7.1. Постановка задачи

Проблема реализации управления возникает в том случае, если при синтезе реализуемого управления U^* не были учтены какие-то важные факторы, влияющие на достижение целей Z^* управления в объекте. К ним относятся, например, состояние X среды, ненаблюдаемые факторы E и отклонение объекта F^0 от его модели F . Причиной неопределенности являются:

- ограниченность всякого прогноза состояния среды. Дело в том, что управление U^* синтезировалось для прогнозируемого состояния X , которое, естественно, неизбежно отличается от действительного;

- влияние ненаблюдаемых факторов E , которые изменяют ситуацию, складывающуюся в процессе управления как в объекте, так и в среде;

- несоответствие между объектом F^0 и его моделью F . Это несоответствие может иметь двоякий характер. С одной стороны, оно может быть вызвано дрейфом параметров объекта за время от наблюдения его состояния Y до момента реализации управления, а с другой — намеренно заложено в модель F . Ради простоты модели часто идут на заведомое игнорирование какой-то информации об объекте, хотя ее и можно было бы получить и использовать при синтезе модели. Расчет здесь прост: так как в процессе реализации управления U^* всегда возникают всякие непредвиденные обстоятельства, то к ним заодно можно отнести и указанное несоответствие, чтобы сразу скорректировать управление с учетом всех имеющихся конкретных обстоятельств. Именно поэтому реализация управления U^* неизбежно связана с его коррекцией, учитывающей условия, кото-

рые сложились к данному моменту. Новое управление имеет вид

$$(U^*)' = U^* + \delta U^*,$$

где δU^* — коррекция управления, учитывающая указанные выше непредвиденные изменения в среде и объекте.

Рассмотрим два вида непредвиденных изменений среды и объекта и способы их учета при коррекции управления.

7.2. Учет влияния среды

Статический объект. Пусть оптимальное управление U^* синтезировано для определенного прогнозируемого состояния среды $\hat{X}$. В действительности реализовался не прогноз $\hat{X}$, а $X \neq \hat{X}$. Информация о X позволяет скорректировать управление, если известно, как должен реагировать объект на управление U . Пусть эта реакция Y^* . Тогда действительная реакция $Y = F(X, U^*)$ будет отличаться от требуемой Y^* на величину

$$\delta Y^* = F(X, U^*) - Y^*.$$

Здесь F — функция (так как рассматривается модель статического объекта).

Очевидно, что коррекция управления δU должна минимизировать эту величину, т. е. для синтеза коррекции δU необходимо решить минимизационную задачу:

$$Q(\delta U) = |F(X, U^* + \delta U) - Y^*| \rightarrow \min_{\delta U \in R'}, \quad (7.2.1)$$

где R' — ресурсы, выделяемые на коррекцию.

Воспользуемся малостью δU и δX — отклонения реального состояния среды X от прогнозируемого $\hat{X}$: $\delta X = X - \hat{X}$. Тогда минимизируемую функцию $Q(\cdot)$ можно записать в виде линейного разложения

$$Q(\delta U) = \left| F(\hat{X}, U^*) + \frac{\partial F}{\partial X} \delta X + \frac{\partial F}{\partial U} \delta U - Y^* \right|$$

и, предположив для примера, что $n=m=q=1$ и ресурсы R' неограничены, получить из условия $Q(\delta U) = 0$ в явном виде для коррекции

$$\delta u = - \left(\frac{\partial f}{\partial u} \right)^{-1} \left[f(\hat{x}, u^*) + \frac{\partial f}{\partial x} \delta x - y^* \right],$$

где значения производных функции $f(\cdot, \cdot)$ вычисляются при u^* и $\hat{x}$.

Динамический объект. При коррекции управления в процессе его реализации на динамическом объекте следует минимизировать невязку между требуемым изменением выхода объекта $Y^*(t)$ в процессе управления и тем, что получается при управлении в среде $X(t) \neq \hat{X}(t)$, где $\hat{X}(t)$ — прогноз поведения среды. Эта задача имеет вид

$$\int_0^T [F(X(t), U^*(t) + \delta U(t)) - Y^*(t)]^2 dt \rightarrow \min_{\delta U(t) \in R'},$$

где F — оператор динамической модели. Здесь, однако, необходимо предвидеть поведение среды $X(t)$, что можно реализовать только путем прогнозирования, однако более позднего, чем $\hat{X}(t)$, с помощью которого было определено управление $U^*(t)$, и, следовательно, более точного.

7.3. Учет активности объекта

Если объект оказался активным (как например, учреждение, цех, завод и т. д.) и это обстоятельство не учитывалось на стадии синтеза управления, то при его реализации без учета активности объекта обойтись уже нельзя. Покажем на простом примере, какие проблемы здесь возникают.

Дело в том, состояние Y активного объекта зависит не только от среды X и управления U , но и от некоторых его внутренних параметров $W = (w_1, \dots, w_l)$, которые он может изменять по своему усмотрению в соответствии со своими собственными целями Z^*_0 (в этом и состоит активность объекта). Эти цели редко совпадают с целями Z^* управления, что и создает трудности при управлении таким активным объектом.

Таким образом, состояние активного объекта можно представить в виде $Y = F_0(X, U, W)$, где F_0 — оператор объекта.

Пусть цели Z^*_0 и Z^* экстремальные. Тогда задача, решаемая при синтезе управления U , имеет вид

$$Z^* : Q_u(U, W) \rightarrow \min_{U \in R_u} \Rightarrow U^*_w, \quad (7.3.1)$$

где $Q_u(\dots)$ — целевая функция управления; R_u — ресурс, выделяемый на управление; U^*_w — принятое решение, которое, естественно, зависит от того, какие значения параметров W объект выберет. (Состоянием среды для простоты записи пренебрегаем, т. е. считаем среду неизменяющейся.) Объект же в силу своей активности, добиваясь своих целей Z^*_0 , решает иную задачу:

$$Z^*_0: Q_w(U, W) \rightarrow \min_{W \in R_w} \Rightarrow W^*_u. \quad (7.3.2)$$

где $Q_w(\dots)$ — целевая функция объекта; R_w — ресурс, которым располагает объект; W^*_u — решение, принятое объектом, которое зависит от решения U . (Как видно, оба решения U^*_w и W^*_u зависят друг от друга. Эта зависимость может прежде всего привести к неустойчивости ситуации, что определяется оператором F_0 объекта и функциями цели Q_{u^*} и Q_{w^*} , но не это главное.)

Пусть решения W^*_u и U^*_w устойчивые. Естественно задать вопрос: а можно ли всё-таки улучшить сложившуюся ситуацию, т. е. уменьшить значение целевой функции Q_u при принятых решениях U^*_w и W^*_u ? Оказывается, можно. Для этого следует рассмотреть управление как процесс взаимодействия двух равных партнеров, т. е. учитывать, что противоположная сторона также добивается экстремальных целей. Учет этого обстоятельства обеими сторонами (объектом и управляющим устройством) позволяет построить такое компромиссное управление, которое обеим сторонам обеспечивает наиболее благоприятные условия, чем дают U^*_w и W^*_u , полученные при решении задач (7.3.1) и (7.3.2) без учета активности сторон.

Проиллюстрируем сказанное на простом примере. Пусть

$$Q_w = u^2 + w^2, \quad Q_u = (u-1)^2 + (w-1)^2.$$

Рельеф обеих функций показан на рис. 7.3.1,а. Это концентрические окружности с центрами (0,0) и (1,1). Решение задач (7.3.1) и (7.3.2)

$$Q_u = (u-1)^2 + (w-1)^2 \rightarrow \min_u \Rightarrow u^* = 1,$$

$$Q_w = u^2 + w^2 \rightarrow \min_w \Rightarrow w^* = 0$$

приводит к устойчивой точке $\mathbf{B}=(1,0)$, которая обеспечивает партнерам единичные значения их целевых функций:

$$Q_u(1,0)=Q_w(1,0)=1. \quad (*)$$

Это все, чего можно достичь, не учитывая разумности партнеров. Но, как легко заметить на рис. 7.3.1,а, на плоскости существует зона, где обеим сторонам обеспечиваются меньшие значения их целевых функций (на рисунке она заштрихована). Ввиду осевой симметрии обеих целевых функций их поведение можно рассмотреть лишь на прямой OA , соединяющей обе цели. Этот разрез показан на рис. 7.3.1,б. Хорошо видно, что заштрихованная зона более выгодна обеим сторонам, чем точка B . Так, если они договорятся поддерживать состояние в компромиссной точке $C=(1/2, 1/2)$, то их

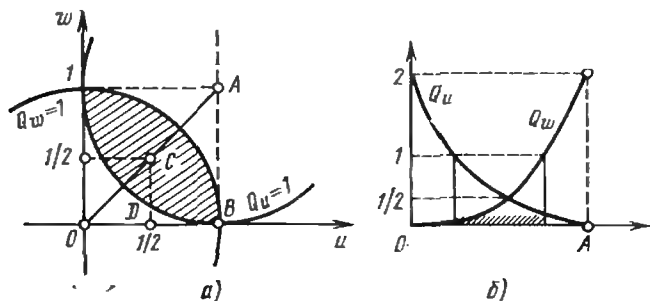


Рис. 7.3.1. Пример «рельефа» целевых функций (а) и показателей качества вдоль оси AO (б)

показатели качества будут равны $Q_u(1/2, 1/2)=Q_w(1/2, 1/2)=1/2$, т. е. обе стороны имеют двукратный выигрыш от такого решения по сравнению с точкой B . Полученный эффект зависит от структуры целевых функций обеих сторон. Что же нужно сделать, чтобы реализовать этот эффект?

В данном конкретном случае будет достаточно, если каждая сторона сообщит значение своей целевой функции другой стороне и они договорятся минимизировать каждый своими средствами сумму:

$$Q_w(u, w) + Q_u(u, w) \rightarrow \min_u \Rightarrow u^* = 1/2,$$

$$Q_w(u, w) + Q_u(u, w) \rightarrow \min_w \Rightarrow w^* = 1/2.$$

Полученное решение, однако, неустойчиво. Действительно, достаточно одной стороне (например, объекту) нарушить уговор и снова перейти к управлению по своей целевой функции (7.3.2), то ситуация сместится в точку $D=(1/2, 0)$, где объект имеет $Q_w(1/2, 0)=1/4$, т. е. для него эта ситуация на $1/4$ лучше, чем договорная. Однако значение целевой функции управляющего устройства при этом ухудшится на $3/4$: $Q_u(1/2, 0)=5/4$ и оно вправе перейти к управлению по своему критерию Q_u , что возвращает систему в устойчивую точку $\mathbf{B}=(1, 0)$. Как видно, указанный договор является неустойчивым,

но полезным состоянием взаимодействия для обеих сторон в процессе управления.

Таким образом, при реализации управления активным объектом следует обязательно учитывать его интересы и использовать договорные отношения для того, чтобы удовлетворять не только цели управления, но и цели объекта, что обеспечивает эффективную реализацию целей Z^* .

ЛИТЕРАТУРНЫЙ КОММЕНТАРИЙ К ГЛАВЕ 7

Задачу реализации управления можно подразделить на две — оперативную коррекцию управления в связи с возникновением новой ситуации в среде и объекте и собственно реализацию этого управления. Вторая задача связана с изменением положения органов управления объектом $u_1, \dots, u_q$ исполнительными механизмами в соответствии с оптимальной программой $U^*(t)$, полученной на стадии синтеза управления. Автоматическое отслеживание этой программы производится с использованием хорошо разработанных методов теории автоматического управления. Есть много хороших руководств по этому вопросу, отметим некоторые из них [1—10].

Активность объекта, рассмотренную в § 7.3, строго говоря, следует учитывать уже на стадии синтеза его модели. Это означает, что необходимо заранее знать цели активного объекта и его возможности по реализации этих целей. Формализация этой информации приводит к обычной модели $F(X, U)$, синтез которой рассмотрен в гл. 3—5. Некоторые способы формализации такого рода информации рассмотрены в [11].

Если же субъект не располагает такой информацией, то на стадии реализации управления он немедленно обнаруживает результат активности объекта, так как последний оказывается не в том состоянии; на которое рассчитывал субъект, опираясь на свою модель этого объекта. Именно поэтому на стадии реализации управления субъекту приходится «входить в контакт» с объектом, т. е. договариваться о «правилах игры», в которой объекту предоставляются определенные права. Субъект также должен ограничить себя определенными договорными обязательствами с объектом. Теперь учет этих обязательств образует новую цель, в рамках которой и следует строить оптимальное управление.

Этой сложной и конфликтной ситуацией занимается теория игр, в частности, новая ветвь, связанная с неантагонистическими играми. С этим новым направлением можно ознакомиться по брошюре [12] и книге [13].

Активный объект всегда включает в себя человека или коллективы. Такой объект иногда называют эргатическим. Специфика поведения эргатических систем рассмотрена в монографии [14]. Особенности поведения человека как объекта управления исследованы в работах [15, 16].

СПИСОК ЛИТЕРАТУРЫ

1. Теория автоматического управления/ Под ред. А. В. Нетушила. — М.: Высшая школа, 1976.
2. Теория автоматического управления/ Под ред. А. В. Нетушила. — М.: Высшая школа, 1972, Ч. II.

3. Воронов А. А. Основы автоматического управления. — М.: Энергия, 1965. Ч. I. Линейные системы регулирования одной величины.
4. Воронов А. А. Основы автоматического управления. — М.: Энергия, 1966. Ч. II. Специальные линейные и нелинейные системы автоматического регулирования одной величины.
5. Воронов А. А. Основы автоматического управления. — Л.: Энергия, 1970. Ч. III. Оптимальные многосвязные и адаптивные системы.
6. Основы автоматического управления/ Под ред. В. С. Пугачева. — М.: Наука, 1968.
7. Сапошников Р. А. Основы технической кибернетики. — М.: Высшая школа, 1970.
8. Егоров К. В. Основы теории автоматического регулирования. — М.: Энергия, 1967.
9. Юревич Е. И. Теория автоматического управления. — Л.: Энергия, 1969.
10. Бесекерский В. А., Попов Е. П. Теория систем автоматического регулирования. — М.: Наука, 1966.
11. Бурков В. Н. Основы математической теории активных систем. — М.: Наука, 1977.
12. Ватель И. А., Ерешко Ф. И. Математика конфликта и сотрудничества. — М.: Знание, 1973.
13. Гермейер Ю. Б. Игры с противоположными интересами. — М.: Наука, 1976.
14. Павлов В. В. Начало теории эргатических систем. — Киев: Наукова думка, 1975.
15. Акофф Р., Эмери Ф. О целеустремленных системах: Пер. с англ. — М.: Сов. радио, 1974.
16. Лефевр В. А. Конфликтующие структуры. — М.: Сов. радио, 1973.

ГЛАВА 8

КОРРЕКЦИЯ СИСТЕМЫ УПРАВЛЕНИЯ

Необходимость коррекции системы управления возникает в связи с несовпадением реакции объекта $Y^0 = F^0(X, U)$ и реакции его модели $Y = F(X, U)$ в результате реализации управления U в обстановке X , т. е.

$$Y^0 \neq Y. \quad (8.0.1)$$

При наличии помех в процессе измерения состояний объекта или при стохастическом характере его поведения в качестве Y^0 фигурирует его среднее значение. При отсутствии необходимого статистического материала для вычисления среднего, отклонение Y^0 от Y при этом должно быть достоверно, например, выходить за пределы «трех сигм».

Итак, при достоверном выполнении неравенства (8.0.1) необходима коррекция системы управления. Эта коррекция в соответствии с рис. В.4.1 может затрагивать модель, объект и цели управления. В первом случае она может распространяться как на параметры модели, так и на ее структуру. Очевидно, что коррекция структуры происходит реже, чем параметров, которые могут корректироваться на каждом шаге процесса управления.

Если коррекция модели не приводит к ожидаемому успеху, обращаются к коррекции объекта, т. е. к изменению границ, отделяющих его от среды.

Процесс реализации целей Z^* создаваемой системы управления не может не привести к коррекции исходного множества целей. Дело в том, что множество целей, которое достигается реализованной системой управления, как правило, не совпадает с исходным множеством целей. Это связано с тем, что сложный объект эволюционирует каким-то определенным, но неизвестным нам образом и, естественно, не все исходные цели могут быть полностью реализованы. С другой стороны, открываются дополнительные возможности по реализации новых целей. Это и вызывает необходимость коррекции множества целей, т. е. определения действительных возможностей данной системы управления на данном объекте.

Рассмотрим все три упомянутых вида коррекции.

8.1. Коррекция параметров модели объекта

Пусть модель объекта имеет вид

$$Y = F(X, U, C), \quad (8.1.1)$$

где F , как обычно, структура модели, а $C = (c_1, \dots, c_h)$ — ее параметры. В процессе управления изменяются оба входа объекта X и U , что и дает возможность информацию о работе объекта на N -м шаге управления представить в виде тройки замеров состояний:

$$I_N = \langle X_N, U_N, Y_N^o \rangle, \quad (8.1.2)$$

где $Y_N^o = F^o(X_N, U_N)$.

Для статического объекта эта тройка представляет собой набор чисел, а для динамического — набор функций:

$$I_N(t) = \langle X_N(t), U_N(t), Y_N^o(t) \rangle. \quad (8.1.3)$$

Рассмотрим коррекцию параметров модели статиче-

ского и динамического объектов для случая, когда его структура совпадает со структурой модели, т. е.

$$\mathbf{F}=\mathbf{F}^0, \quad (8.1.4)$$

а параметры объекта каким-то неизвестным, но определенным образом изменяются во времени, т. е.

$$\mathbf{C}^0=\mathbf{C}^0(t), \quad (8.1.5)$$

где $\mathbf{C}^0(t)$ — неизвестная вектор-функция дрейфа параметров объекта.

Задачей коррекции параметров является такое изменение параметров модели $\mathbf{C}$, которое отражало бы изменение параметров объекта $\mathbf{C}^0(t)$, т. е.

$$\mathbf{C} \rightarrow \mathbf{C}^0(t). \quad (8.1.6)$$

Алгоритм решения этой задачи назовем алгоритмом коррекции. Он должен учитывать информацию о работе объекта и иметь рекуррентный характер, т. е.

$$\mathbf{C}_{N+1}=\mathbf{C}_N+\varphi(\mathbf{C}_N, I_N), \quad (8.1.7)$$

где φ — алгоритм коррекции. Заметим, что рекуррентный характер алгоритма коррекции связан с самой постановкой задачи коррекции. Это обстоятельство дает возможность при коррекции модели воспользоваться так называемым методом самонастраивающейся модели. Схема реализации этого метода показана на рис. 8.1.1. Здесь система управления (она обозначена пунктиром) реализует задаваемые ей извне цели $\mathbf{Z}^*$. Информация о поведении объекта сопоставляется с реакцией модели $\mathbf{F}$ на входы $\mathbf{X}$ и $\mathbf{U}$ и разность $\Delta \mathbf{Y}_N=\mathbf{Y}_N-\mathbf{Y}^0$ поступает на блок коррекции, который и производит коррекцию параметров модели:

$$\Delta \mathbf{C}_{N+1}=\varphi(\mathbf{C}_N, \mathbf{X}_N, \mathbf{U}_N, \Delta \mathbf{Y}_N). \quad (8.1.8)$$

Так как модель $\mathbf{F}$ при векторном выходе $\mathbf{Y}=(y_1, \dots, y_m)$ может быть представлена в виде m моделей $\mathbf{F}=(f_1, \dots, f_m)$, то коррекция должна иметь место для каждой модели

$$y_i=f_i(\mathbf{X}, \mathbf{U}, \mathbf{C}^i), \quad i=1, \dots, m,$$

где y_i — выход i -й модели; f_i — ее структура, а $\mathbf{C}^i$ — ее параметры. Как показано в § 3.3, в данном случае можно рассматривать коррекцию одной модели, подразумевая, что коррекция остальных будет производиться аналогично.

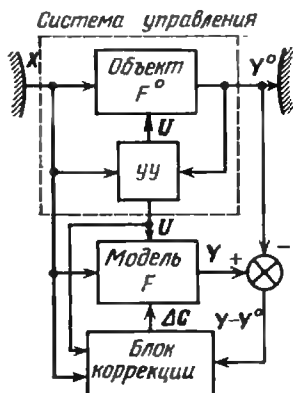
Поэтому ниже мы будем рассматривать коррекцию модели вида

$$y=f(X, U, C) \quad (8.1.9)$$

со скалярным выходом y , где f — известный оператор модели.

Как легко заметить, процесс коррекции параметров модели очень похож на процесс адаптивной идентификации, рассмотренной в гл. 4. Используем эту аналогию для синтеза алгоритмов коррекции.

Рис. 8.1.1. Схема коррекции параметров модели методом самонастраивающейся модели



Рассмотрим алгоритмы Φ коррекции для статического и динамического объектов.

Статический объект. Он характеризуется статической структурой, которая описывается заданной скалярной функцией f [см. (8.1.9)]. Используя аналогию с адаптивной идентификацией, будем искать алгоритм коррекции среди алгоритмов минимизации локальной невязки выходов модели и объекта в момент N :

$$q^2_N = (y_N - y^o_N)^2. \quad (8.1.10)$$

Подставляя сюда выражение для выхода модели (8.1.9), получаем эту невязку в виде функции от параметров C_N модели

$$q^2_N(C_N) = (f(X_N, U_N, C_N) - y^o_N)^2. \quad (8.1.11)$$

Минимизируем эту невязку, изменяя (корректируя) параметры модели, т. е. решая задачу

$$q^2_N(C_N + \Delta C_{N+1}) \rightarrow \min_{\Delta C_{N+1}}.$$

Здесь естественно воспользоваться коррекцией, которая при минимальном изменении C_N обеспечивает наибольший эффект уменьшения невязки q^2 . Такой коррекцией будет градиентная коррекция

$$\Delta C_{N+1} = -\alpha \nabla_C q^2_N(C_N), \quad (8.1.12)$$

где α — положительный параметр (о нем будет сказано ниже); $\nabla_C q^2_N$ — градиент по C функции невязки (8.1.11) при $C = C_N$, который по определению равен

$$\nabla_C q^2_N = 2q_N \nabla_C f(X_N, U_N, C)|_{C=C_N}, \quad (8.1.13)$$

где

$$\nabla_C f(X_N, U_N, C) = \left(\frac{\partial f(X_N, U_N, C)}{\partial c_1}, \dots, \frac{\partial f(X_N, U_N, C)}{\partial c_k} \right).$$

Эти выражения легко вычисляются при аналитически заданной структуре функции f модели. Если эта функция очень сложна или задана алгоритмически, например для имитационной модели, ее производные могут быть оценены следующим образом:

$$\frac{\partial f}{\partial c_i} = \frac{1}{2g_i} [f(X_N, U_N, C_N + e_i g_i) - f(X_N, U_N, C_N - e_i g_i)], \quad (8.1.14)$$

где g_i — база оценки; e_i — i -й орт.

Теперь выражение для коррекции можно записать в виде

$$\Delta C_{N+1} = -2\alpha [f(X_N, U_N, C_N) - y^0_N] \nabla_C f(X_N, U_N, C)|_{C=C_N}. \quad (8.1.15)$$

Рассмотрим вопрос о выборе параметра α . При отсутствии помех он должен назначаться из соображений наиболее полной минимизации невязки. А так как ее минимальное значение не может быть меньше нуля, то выбор α связан с решением задачи, рассмотренной в гл. 4:

$$q^2_N(C_N + \Delta C_{N+1}) \rightarrow \min_{\alpha} = 0 \Rightarrow \alpha^* \quad (8.1.16)$$

(см. рис. 4.2.1). Эта задача однопараметрической минимизации, которая может быть решена любым из известных поисковых методов (например, методом золотого сечения, описанным в приложении П. 2).

Если же исходная информация о поведении объекта «зашумлена», то параметр α должен быть меньше α^* . Действительно, величина коррекции ΔC_{N+1} пропорциональна невязке $y_N - y^0_N$, которая при наличии помех значительна не за счет неточности модели, а за счет помех, накладывающихся в процессе измерения состояния объекта. Поэтому параметр α должен уменьшать коррекцию при наличии помех и тем более, чем выше уровень помех.

Таким образом, коррекция параметров статической модели производится по формуле (8.1.15) и связана с вычислением или оценкой градиента функции выхода модели по корректируемым параметрам (8.1.14).

Динамический объект. В этом случае f в (8.1.9) является оператором определенного вида, а C — вектором его параметров. Рассмотрим дискретный динамический объект, оператор которого имеет конечно-разностную форму вида (4.3.4). Модель его при наличии управления записывается в виде

$$y_{N+1} = f(X'_{N+1}, U_{N+1}, Y'_N, C),$$

где $Y'_N = (y_N, y_{N-1}, \dots, y_{N-p})$; f — заданная функция четырех векторных аргументов X' , U , Y' и C , определенная на стадии структурного синтеза.

Пусть реакции модели y_N и объекта y^0_N не совпадают, т. е. $y_N \neq y^0_N$. Это обстоятельство заставляет обращаться к коррекции параметров C модели путем минимизации функции невязки, решая задачу

$$q^2_N(C) = [f(X'_N, U_N, Y'_{N-1}, C) - y^0_N]^2 \rightarrow \min_C.$$

Для этого естественно воспользоваться градиентным методом:

$$C_{N+1} = C_N - \alpha \nabla_C q^2_N(C) |_{C=C_N},$$

где для оценки градиента вида (8.1.13) необходимо иметь оценки производных, аналогичные (8.1.14):

$$\frac{\partial f}{\partial c_i} = \frac{1}{2g_i} [f(X'_N, U_N, Y'_{N-1}, C_N +$$

$$+ e_i g_i) - f(X'_N, U_N, Y'_{N-1}, C_N - e_i g_i)], \quad i = 1, \dots, k.$$

При этом параметр α выбирается так же, как и для статического объекта.

Таким образом, основная трудность коррекции параметров объекта состоит в оценке градиента функции ло-

кальной невязки выходов объекта и модели по корректируемым параметрам. Однако указанная процедура производится только на основе информации, полученной в результате реализованного в объекте управления, и корректирует модель именно для такого управления. Если на следующем этапе понадобится иное управление, которое опирается на неизвестные ранее свойства объекта (а так и бывает в большинстве случаев), то модель не будет «готова» к этому. Это обстоятельство заставляет добавлять к оптимальному управлению $U^*(t)$ так называемую изучающую добавку $\delta U(t)$, которая, не слишком искажая оптимальное управление, позволяла бы более качественно корректировать модель. По сути дела речь идет о планировании экспериментов в процессе управления. Добавку $\delta U(t)$ следует определять таким образом, чтобы произведенная на ее основе коррекция параметров модели ΔC позволила бы создать более точную модель.

Как видно, в этом случае суммарное управление $U^*(t) + \delta U(t)$ преследует две цели — добиться заданной цели управления Z^* и одновременно подстроить модель объекта F . Очевидно, что в этом случае управление должно быть оптимальным. Такое управление, преследующее двойную цель, получило название дуального, т. е. двойственного. Его методы связаны с решением описанной задачи, они очень сложны и поэтому не будут здесь излагаться. Суть же их заключается в минимизации некоторой функции, определяющей суммарный ущерб от недостижения цели управления (за счет присутствия добавки $\delta U(t)$) и неточности модели. Минимизация этих потерь и дает возможность определить оптимальное дуальное управление.

8.2. Коррекция структуры модели

Коррекцию структуры модели рассмотрим как переход с одной структуры к другой в процессе управления.

Пусть известно, что изменение структуры объекта происходит в пределах конечного числа заданных структур (функций или операторов) моделей: $f^1, \dots, f^k$. Это означает, что в каждый момент времени одна из этих моделей

$$y^i = f^i(X, U, C^i), \quad i=1, \dots, k \quad (8.2.1)$$

наилучшим образом описывает изменяющуюся структуру объекта

$$y^0_t = f^0_t(\mathbf{X}, \mathbf{U}),$$

где $\mathbf{C}^i$ — вектор параметров i -й модели; f^0_t — структура объекта, которая изменяется каким-то неизвестным образом во времени t .

В качестве критерия эффективности структуры естественно выбрать каким-то образом усредненную невязку каждой модели и объекта, вычисленную до коррекции параметров (иначе невязки будут равны нулю или близки к нему). Пусть для статического объекта

$$Q^i_j = [f^i_j(\mathbf{X}, \mathbf{U}, \mathbf{C}^i_j) - y^0_j]^2$$

— невязка i -й модели на j -м шаге управления до коррекции ее параметров $\mathbf{C}^i$. Однако, полученные таким образом значения невязок на j -м шаге управления $Q^i_j, \dots, Q^k_j$ мало что говорят об эффективности каждой из k моделей, так как на эти величины влияют всякого рода случайные факторы. Поэтому естественно ввести показатели эффективности структур в виде взвешенных сумм этих невязок:

$$\bar{Q}^i_N = \sum_{j=1}^N g^{N-j} Q^i_j, \quad i = 1, \dots, k, \quad (8.2.2)$$

где $0 \leq g < 1$ — коэффициент «забывания», который как бы «обесценивает» информацию, полученную на предыдущих шагах (чем дальше «в прошлое», тем меньшую роль играют измерения). При $g=0$ получим $\bar{Q}^i_N = Q^i_N$, т. е. взвешенная невязка равна невязке на последнем шаге, так как при этом предыстория вся «забыта».

Можно обобщить выражение для показателей эффективности (8.2.2), заменив весовой коэффициент g^{N-j} функцией $\varphi(j)$:

$$\bar{Q}^i_N = \sum_{j=1}^N \varphi(N-j) Q^i_j, \quad i = 1, \dots, k.$$

Здесь $\varphi(\cdot)$ — монотонно убывающая функция «забывания», обладающая следующими свойствами:

$$\begin{aligned} \Xi: \varphi(i) > 0, \quad \varphi(0) = 1, \quad \lim_{i \rightarrow \infty} \varphi(i) = 0, \\ \varphi(i) \geq \varphi(i+1) \quad (i = 0, 1, \dots). \end{aligned} \quad (8.2.3)$$

Теперь довольно легко выработать решающее правило. Очевидно, что наиболее подходит та модель, показатель эффективности которой минимален. Это означает, что на N -м шаге решение о выборе структуры принимается следующим образом:

$$f_N = f', \text{ если } \bar{Q}'_N = \min_{l=1, \dots, k} \bar{Q}^l_N. \quad (8.2.4)$$

Таким образом, процедура адаптации структуры сводится к оценке на каждом шаге показателя эффективности для всех конкурирующих структур и выбору той, у которой этот показатель минимален. Как видно, для реализации указанной процедуры необходимо иметь и постоянно корректировать параметры k моделей, из которых лишь наилучшая используется для синтеза управления.

При наличии помех особую роль играет выбор функции забывания $\varphi(\cdot)$. Чем выше уровень помех в объекте, тем медленнее должна убывать эта функция (тем самым осуществляется фильтрация). Однако, с другой стороны, при интенсивном изменении структуры объекта эта функция должна уменьшаться достаточно быстро, чтобы быстрее «забыть» результаты предыдущих экспериментов, уже играющих на данный момент малую роль ввиду быстрого изменения структуры объекта. Поэтому для синтеза оптимального решающего правила (8.2.4) важно правильно выбрать функцию $\varphi(\cdot)$. Как это сделать?

Для этого необходимо оценить вероятность принятия ошибочного решения P , т. е. решения о выборе l -й структуры, в то время как в действительности лучшей оказывается другая структура. Эта вероятность зависит от функции $\varphi(\cdot)$. Теперь задачу оптимизации решающего правила (8.2.4) следует представить как задачу минимизации этого функционала вероятности ошибки:

$$P(\varphi(i)) \rightarrow \min_{\varphi(i) \in \Xi} \Rightarrow \varphi^*(i), \quad (8.2.5)$$

где Ξ — множество допустимых функций, обладающих свойствами (8.2.3); $\varphi^*(i)$ — оптимальная функция «забывания».

Для определения вероятности P ошибки необходимо знать интенсивность и характер дрейфа структуры объекта, т. е. его вероятностные свойства.

8.3. Коррекция объекта управления

Эволюция сложного объекта F^0 в процессе управления, дрейф его параметров и характеристик, изменение окружающей его среды, т. е. множества состояний $\{X\}$, а также множества целей $\{Z^*\}$ может привести к тому, что несмотря на адаптацию параметров и структуры модели объекта система управления не будет обеспечивать эффективное выполнение всех заданных целей, которые должны быть реализованы в объекте. Это обстоятельство заставляет пересмотреть границу разделения объекта и среды, т. е. корректировать объект управления, первоначальный вариант которого был определен на этапе выделения объекта из среды (см. гл. 2).

Для коррекции объекта в процессе управления необходимо иметь альтернативы, т. е. варианты объекта, из которых можно осуществлять выбор при коррекции. Пусть число таких альтернатив ограничено и ими являются

$$F^{01}, F^{02}, \dots, F^{0g}. \quad (8.3.1)$$

Для каждого варианта объекта из (8.3.1) необходимо иметь его модель, которая синтезируется методами, рассмотренными выше (см. гл. 3—5):

$$Y^i = F^i(X^i, U^i), \quad i=1, \dots, g, \quad (8.3.2)$$

где F^i — оператор i -й модели объекта F^{0i} ; X^i — контролируемые, а U^i — управляемые входы этого объекта. Эти входы отличаются как по числу, так и по содержанию. Выходы же всех объектов Y^i имеют одинаковое содержание для всех объектов, но различны по своим значениям. Оценку эффективности объекта естественно производить по степени достижения им поставленной цели Z^* управления. Для этого необходимо ввести меру недостижения цели в виде скалярной функции двух переменных

$$v = v(Z^*, \Psi(Y)), \quad (8.3.3)$$

где $\Psi(\cdot)$ — преобразование пространства состояний $\{Y\}$ объекта в целевое пространство $\{Z\}$ [см. (1.3.6)].

Для целей типа равенств эту функцию можно определить, например, так: $v = |Z^* - \Psi(Y)|^2$. Несколько сложнее это сделать для целей (ограничений) типа неравенств и трудно, но возможно для экстремальных це-

лей. Так или иначе, но считаем, что функция (8.3.3) определена и имеется возможность численно оценить степень реализации заданной цели в объекте.

Предположим далее для простоты, что цель Z^* и состояние X среды неизменны во времени (это задача стабилизации объекта в заданном состоянии $Y^* = \Psi^{-1}(Z^*)$, где Ψ^{-1} — функция, обратная Ψ) и нет случайных помех. Тогда решение задачи выбора лучшего из (8.3.1) объекта заключается в последовательном переборе и управлении каждым из g конкурирующих объектов и фиксации значений

$$v^i = v(Z^*, \Psi(Y^i)), \quad i=1, \dots, g.$$

Заканчивается эта процедура выбором такого объекта, который обеспечивает наименьшее значение величины v . Это правило записывается в простом виде:

$$F^0 = F^{0^i}, \text{ если } v^i = \min_{i=1, \dots, g} v^i. \quad (8.3.4)$$

Но подобный перебор нецелесообразен при быстром изменении свойств объекта, так как и «плохой» и «хороший» объекты будут включаться одинаково (точнее, почти одинаково) часто. В этом случае целесообразно обратиться к рандомизированному правилу. Для этого следует определить вероятности выбора того или иного объекта:

$$p^1, \dots, p^g \left(\sum_{i=1}^g p^i = 1 \right), \quad (8.3.5)$$

где p^i — вероятность выбора i -го объекта. Эти вероятности можно связать со значением показателя эффективности:

$$p^i = f(v^i), \quad i=1, \dots, g,$$

где f — обратно пропорциональная функция вида $f(v) = a/(v+b)$, причем параметр a является нормирующим множителем для выполнения условия (8.3.5) на каждом шаге, а параметр b численно должен быть равным оценке среднего значения эффективности всех объектов:

$$b = \frac{1}{g} \sum_{i=1}^g v^i$$

и поэтому должен адаптироваться в процессе управления.

В начальный момент $v^i=0$ ($i=1, \dots, g$). Далее, по мере работы того или иного варианта объекта вероятности (8.3.5) будут принимать значения

$$p^i = a / (v^i + b), \quad (8.3.6)$$

где v^i — эффективность последнего функционирования i -го варианта объекта. Как видно, в этом случае чаще будет включаться лучший объект, что и обеспечивает коррекцию объекта, т. е. адаптацию его границ к изменяющимся свойствам объекта.

Теперь обобщим сказанное и снимем ограничения на состояние X среды, цель Z^* и наличие помех. Эти факторы, изменяясь непредвиденным образом, сделают каждый результат оценки v стохастическим, что заставляет обращаться к процедурам фильтрации типа

$$\bar{v}_{N+1}^i = (1 - \alpha) \bar{v}_N^i + \alpha v_{N+1}^i, \quad (8.3.7)$$

где α — коэффициент забывания ($0 < \alpha \leq 1$); $\bar{v}_{N+1}^i$ — усредненное значение эффективности i -го варианта объекта на $(N+1)$ -м шаге его использования; v_{N+1}^i — оценка эффективности этого объекта по (8.3.3) на $(N+1)$ -м шаге его использования.

Сама процедура принятия решения о выборе объекта при этом не изменяется. Она рандомизирована, причем вероятности альтернатив (8.3.5) при этом равны (8.3.6), где, однако, вместо v^i необходимо подставить (8.3.7). Как легко видеть, указанная процедура при большем «забывании» ($\alpha=1$) вырождается в процедуру, рассмотренную выше, когда вероятности p^i определяются только последней оценкой эффективности v^i .

8.4. Коррекция целей управления

Пусть Λ множество целей $\{Z\}^*$, которые должна реализовать созданная система управления. Это означает, что она создавалась так, чтобы для любой цели $Z^* \in \Lambda$ и состояния среды $X \in S$ находилось управление, переводящее объект в состояние Y , соответствующее реализации цели Z^* . Однако приближенность модели, дрейф характеристик объекта и ограниченные возможности системы коррекции управления при синтезе и реализации управления приводят к тому, что множество реализуемых целей часто весьма существенно отличается от Λ . Кроме того, эволюция объекта управления и изменение

свойств среды приводит к дрейфу множества реализуемых целей. Очевидно, что границы этого множества должны интересовать потребителя системы управления, так как они определяют ее предельные возможности.

Рассмотрим процедуру коррекции границы множества Λ в процессе эксплуатации системы управления. Прежде всего определим, что влияет на реализуемость цели Z^* . Реализуемость ее зависит от характера самой цели и состояния X среды, в которой находится объект управления. Таким образом, пара

$$L = \langle X, Z^* \rangle \quad (8.4.1)$$

определяет успех или неуспех управления. Поэтому границу реализуемости можно очертить с помощью решающей функции

$$R = R(L), \quad (8.4.2)$$

принимавшей два значения (0 и 1). Если $R(L) = 0$, то это значит, что цель Z^* при состоянии X среды не реализуется данной системой управления. Если же $R(L) = 1$, то цель Z^* реализуется данной системой управления при состоянии X среды.

Таким образом, возможности системы управления можно характеризовать функцией (8.4.2). Эволюция объекта и среды, влияющая на реализуемость целей, должна изменять и эту решающую функцию.

В связи с этим возникают две задачи:

— как построить такую решающую функцию для данной системы управления;

— как корректировать эту функцию с изменением (эволюцией) объекта и среды (а может быть, и самой системы управления).

Начнем с решения первой задачи. Для этого необходимо иметь исходную информацию об опыте реализации каких-то целей управления при каких-то состояниях среды, т. е. так называемую обучающую последовательность, образуемую парами соответствий

$$L_i \rightarrow R_i, \quad i = 1, \dots, N, \quad (8.4.3)$$

где L_i i -я пара (8.4.1), а R_i — результат работы системы управления с этой парой (при $R_i = 0$ цель Z^*_i не достигнута, а при $R_i = 1$ достигнута). Располагая этой информацией, необходимо построить соответствующую решающую функцию $R(L)$, принимающую два значения (0 и 1). Эту задачу называют задачей распознавания

образов и состоит она в построении такого решающего правила (распознающей функции) по известной обучающей последовательности (8.4.3).

Существует много методов построения решающего правила для задач распознавания образов (см. комментарий к § 8.4). Для решения данной задачи наиболее подходит так называемый метод потенциальных функций. Рассмотрим его. Прежде всего, образуем пространство $\{L\}$, в котором каждая пара (8.3.1) представляется в виде точки. Координатами этого пространства являются контролируемые параметры среды X и цель Z^* . Таким образом, последовательность (8.4.3) представляется в пространстве $\{L\}$ в виде точек двоякого рода. Одни из них ($R=0$) отражают негативный опыт применения данной системы управления, а другие ($R=1$) — положительный.

Задачи синтеза решающего правила $R(L)$ заключаются в том, чтобы в пространстве $\{L\}$ провести поверхность, разделяющую точки с индексами $R=0$ и $R=1$. Пусть каждая точка L_i образует в пространстве $\{L\}$ «потенциал»

$$\psi_i(L) = \frac{q_i}{1 + v_i \rho(L, L_i)}, \quad (8.4.4)$$

где q_i — «заряд» в точке L_i , вызывающий этот потенциал:

$$q_i = \begin{cases} +1, & \text{если } R_i = 1, \\ -1, & \text{если } R_i = 0; \end{cases} \quad (8.4.5)$$

$v_i > 0$ — некоторая постоянная затухания потенциала; $\rho(L, L_i)$ — функция расстояния между точками L и L_i , например,

$$\rho(L, L_i) = |L - L_i|. \quad (8.4.6)$$

На рис. 8.4.1 показаны потенциалы (8.3.4) при различных знаках заряда. Суммарный потенциал

$$\Psi(L) = \sum_{i=1}^N \psi_i(L) \quad (8.4.7)$$

определяет знак заряда в любой другой точке прост-

ранства и тем самым реализуемость управления в этой точке

$$R(L) = \begin{cases} 1 & \text{при } \Psi(L) \geq 0, \\ 0 & \text{при } \Psi(L) < 0 \end{cases} \quad (8.4.8)$$

или $R(L) = 1/2[1 - \text{sign } \Psi(L)]$, где sign функция знака:

$$\text{sign } z = \begin{cases} +1 & \text{при } z \geq 0, \\ -1 & \text{при } z < 0. \end{cases}$$

Это и есть решающая функция, которая однозначно определяет реализуемость целей управления по наблю-

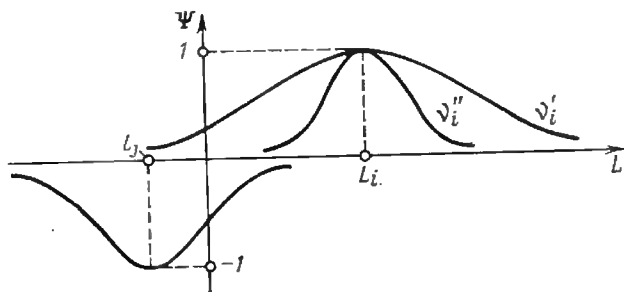


Рис. 8.4.1. Пример потенциалов, вызываемых положительным (L_i) и отрицательным (L_j) «зарядами» ($v''_i > v'_i$)

дениям за реализацией некоторого числа (N) конкретных целей Z^*_i в конкретных ситуациях X_i данной системой управления.

Теперь рассмотрим коррекцию решающей функции (8.4.8). Необходимость такой коррекции связана со «старением» информации (8.4.3), ее неизбежным обесцениванием. Учет такого старения очень просто произвести, используя функцию «забывания» $\varphi(\cdot)$, которая была введена ранее (8.2.3). Для этого достаточно величину потенциала q_i в отличие от (8.4.5) уменьшать следующим образом:

$$q_i = \begin{cases} +\varphi(N-i) & \text{при } R_i = 1, \\ -\varphi_i(N-i) & \text{при } R_i = 0. \end{cases} \quad (8.4.9)$$

При этом номер i должен обязательно отражать историю поступления информации $L_i \rightarrow R_i$, т. е. номера i ($i = 1, \dots, N$) соответствуют дискретным моментам време-

ни t_i реализации цели управления Z^* ; в обстановке X_i , причем $t_i < t_{i+1}$.

В этом случае суммарный потенциал (8.4.7) будет отражать в большей степени «свежую» информацию о работе системы управления и благодаря введению функции (8.4.9) «забывать» старую. Интенсивность этого «забывания» должна зависеть от скорости эволюции объекта и среды и отражать последнюю. Например, функция вида

$$\varphi(i) = \begin{cases} 1 & \text{при } 0 \leq i \leq N_1, \\ 0 & \text{при } N_1 < i \end{cases}$$

позволяет учитывать лишь N_1 последних членов последовательности (8.4.3). Чем меньше N_1 , тем интенсивнее забывание. И, наоборот, при $N_1 = N$ учитывается в равной мере вся имеющаяся информация (8.3.3) и получаем случай (8.4.5).

ЛИТЕРАТУРНЫЙ КОММЕНТАРИЙ К ГЛАВЕ 8

Коррекция системы управления является формой ее адаптации к изменяющимся факторам среды и потребностям субъекта, т. е. к ее новому состоянию и новым целям, реализуемым системой управления. Адаптация эта имеет параметрический и структурный характер. В первом случае изменяются параметры системы управления (например, параметры модели объекта, см. § 8.1), а во втором — ее структура (см., например, § 8.2).

Математический аппарат, используемый для параметрической коррекции в § 8.1, является по сути дела аппаратом адаптивной идентификации, описанный с соответствующими ссылками в гл. 4.

Структурная адаптация, рассмотренная в § 8.2, 8.3, имеет альтернативный характер, т. е. дает возможность на каждом шаге определять одну из нескольких заранее оговоренных альтернатив (моделей в § 8.2 и объектов в § 8.3). Здесь математическим аппаратом является фильтрация (сглаживание) случайных процессов. С этим аппаратом можно познакомиться по любому руководству о случайных процессах, например [1—4].

Изложенный в § 8.4 способ построения решающей функции называется методом потенциальных функций [5]. Однако для этих целей, вообще говоря, могут быть применены и другие методы распознавания образов. Для начального ознакомления с проблемой распознавания образов читателю рекомендуется одна из книг [6, 7]. Более подробно с различными методами распознавания можно ознакомиться по руководствам [5, 8—10].

СПИСОК ЛИТЕРАТУРЫ

1. Свешников А. А. Прикладные методы теории случайных функций. — М.: Наука, 1968.
2. Пугачев В. С. Теория случайных функций. — М.: Физматиз, 1962.

3. **Бендат Дж.** Основы теории случайных шумов и ее применения: Пер. с англ. — М.: Наука, 1965.
4. **Тихонов В. И.** Статистическая радиотехника. — М.: Сов. радио, 1966.
5. **Айзерман М. А., Браверман Э. М., Розоноэр Л. И.** Метод потенциальных функций в теории обучающихся машин.
6. **Аркадьев А. Г., Браверман Э. М.** Обучение машины классификации объектов. — М.: Наука, 1971.
7. **Бонгард М. М.** Проблема узнавания. — М.: Наука, 1967.
8. **Вапинк В. Н., Червоиенкис А. Я.** Теория распознавания образов: статистические проблемы обучения. — М.: Наука, 1974.
9. **Персептрон** — система распознавания образов/ Под ред. А. Г. Ивахненко. — Киев: Наукова думка, 1975.
10. **Загоруйко Н. Г.** Методы распознавания и их применения. — М.: Сов. радио, 1972.

МЕТОДЫ ПОИСКОВОЙ ОПТИМИЗАЦИИ

Задача оптимизации является, пожалуй, самой распространенной вычислительной задачей нашего времени. Действительно, практически все проблемы управления сводятся к отысканию оптимума функции или функционала и на каждом этапе управления сложным объектом приходится решать задачу оптимизации в нескольких модификациях. Поэтому так важно владеть методами решения задачи оптимизации. Существует много способов ее решения. Ниже описаны наиболее эффективные из них, которые неоднократно использовались в задачах управления.

П.1. Постановка задачи

Задача оптимизации функции $Q=Q(X)$ многих переменных $X=(x_1, \dots, x_k)$ формируется следующим образом:

$$Q(X) \rightarrow \min_{X \in \Omega} \Rightarrow X^*, \quad (\text{П.1.1})$$

где Ω — множество допустимых значений переменной X , а X^* — решение задачи оптимизации, т. е.

$$Q(X^*) = \min_{X \in \Omega} Q(X),$$

причем $X^* \in \Omega$. Множество Ω обычно задается системой равенств и неравенств в виде

$$\Omega: \begin{cases} g_i(X) = 0, & i = 1, \dots, p < k, \\ h_j(X) \geq 0, & j = 1, \dots, q, \end{cases}$$

где $g_i(X)$ и $h_j(X)$ — заданные скалярные функции. Задание этих функций, так же как и функции $Q(X)$, может быть не только аналитическим, но и алгоритмическим, т. е. с помощью алгоритма, позволяющего вычислять значения этих функций в любой заданной точке X .

Поисковый метод решения задачи (П.1.1) реализуется рекуррентным выражением, определяющим переход от $(N-1)$ -го приближения к N -му:

$$X_N = X_{N-1} + \Delta X_N.$$

Шаг ΔX_N зависит от ситуации в точке X_{N-1} ; $\Delta X_N = \varphi(X_{N-1})$, где φ — алгоритм поиска, определяющий рекуррентный процесс оптимизации. Этот алгоритм почти полностью определяется признаками задачи оптимизации. Рассмотрим основные из них.

1. В зависимости от числа минимизируемых функций, т. е. от размерности аргумента функции $Q(X)$, различают однокритериальные

(см. П.2—П.5) и многокритериальные векторные задачи оптимизации, которые рассмотрены в П.6.

2. В зависимости от того, лежит решение X^* задачи на границе $\bar{\Omega}$ области Ω или нет, различают безусловные ($X^* \notin \bar{\Omega}$, они представлены в П.2—П.4) и условные ($X^* \in \bar{\Omega}$) задачи оптимизации (их решение рассмотрено в П.5).

3. В зависимости от числа локальных экстремумов различают локальные или одноэкстремальные (они обсуждаются в П.2—П.3) и глобальные или многоэкстремальные задачи (рассмотренные в П.4).

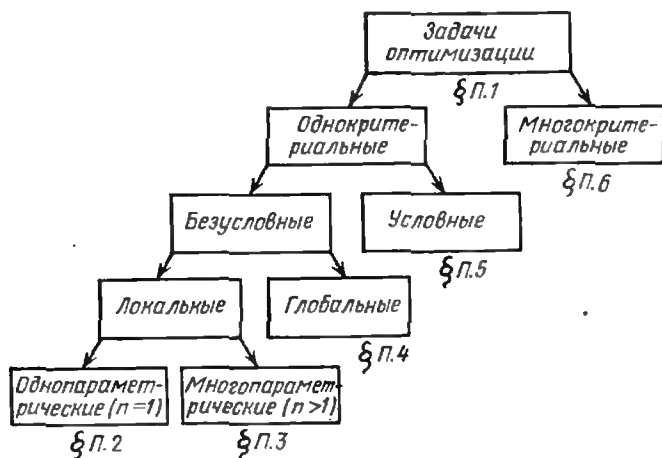


Рис. П.1.1. Структура задач поисковой оптимизации

4. В зависимости от числа n оптимизируемых параметров различают однопараметрические ($n=1$, см. П.2) и многопараметрические ($n>1$ см. П.3) задачи оптимизации.

На рис. П.1.1 показана описанная схема классификации задач поисковой оптимизации с указанием номеров параграфов данного приложения, где рассмотрены основные методы решения этих задач.

П.2. Однопараметрический локальный поиск ($n=1$)

Задача однопараметрического поиска записывается в виде

$$Q(x) \rightarrow \min \Rightarrow x^*. \quad (\text{П.2.1})$$

Если функция $Q(x)$ дифференцируема, то для решения задачи (П.2.1) достаточно решить другую задачу

$$\frac{dQ(x)}{dx} = f(x) = 0. \quad (\text{П.2.2})$$

Если функция не дифференцируема или ее вычисление очень сложно (например, в случае алгоритмического задания функции), то для определения $f(x)$ можно воспользоваться точечной оценкой

$$\hat{f}(x) \approx \frac{dQ(x)}{dx} = \frac{1}{2g} [Q(x+g) - Q(x-g)], \quad (\text{П.2.3})$$

где $g > 0$ — достаточно малая величина (база оценки, равная $2g$).

Как видно из (П.2.2), задача оптимизации сводится к задаче определения корня уравнения $f(x) = 0$. Поэтому для ее решения можно воспользоваться многочисленными рекуррентными методами, разработанными для вычисления корней. Заметим, что ввиду одноэкстремальности функции $Q(x)$ число корней уравнения (П.2.2) не превышает одного.

1. Метод касательных применяют для выпуклой функции $Q(x)$, что соответствует монотонности ее производной $f(x)$. Алгоритм метода на каждом шаге сводится к линейному представлению функции $f(x)$ в точке x_i и решению уравнения

$$f(x) \approx f(x_i) + (x - x_i)f'(x_i) = 0,$$

корень которого и является следующим приближением:

$$x_{i+1} = x_i - f(x_i)/f'(x_i).$$

Как видно, для этого необходимо определить вторую производную $f'(x) = Q''(x)$ от минимизируемой функции. Ее точечная оценка

$$\hat{f}'(x) = \frac{1}{g} \left[f\left(x + \frac{g}{2}\right) - f\left(x - \frac{g}{2}\right) \right] \approx \frac{1}{g^2} [Q(x+g) - 2Q(x) + Q(x-g)],$$

где база оценки равна g .

Таким образом, алгоритм касательных при решении задачи одномерной оптимизации записывается в виде

$$x_{i+1} = x_i - \frac{g [Q(x_i + g) - Q(x_i - g)]}{2[Q(x_i + g) - 2Q(x_i) + Q(x_i - g)]}.$$

Как видно, для того чтобы сделать один шаг оптимизации по методу касательных, необходимо определить минимизируемую функцию в трех точках: x_i , $x_i \pm g$. Напомним, что этот метод сходится, т. е. $\lim_{i \rightarrow \infty} x_i = x^*$, лишь для выпуклой функции $Q(x)$.

2. Метод хорд заключается в том, что функция $f(x)$ в окрестности корня аппроксимируется хордой, соединяющей точки, лежащие по разные стороны от корня. Пусть $x'_i < x''_i$ — две точки в окрестности корня на i -м этапе, причем (для случая минимизации)

$$f(x'_i) < 0, f(x''_i) > 0, \quad (\text{П.2.4})$$

т. е. они лежат по разные стороны от корня x^* . Эта хорда описывается уравнением

$$f(x) = \frac{1}{x''_i - x'_i} [(x - x'_i) f(x''_i) - (x''_i - x) f(x'_i)].$$

Приравнивая это выражение нулю, получаем x_{i+1} — очередное приближение корня. Для очередного шага образуем x'_{i+1} и x''_{i+1} следующим образом:

$$\begin{aligned} x'_{i+1} &= x'_i \text{ и } x''_{i+1} = x_{i+1}, \text{ если } f(x_{i+1}) > 0, \\ x'_{i+1} &= x_{i+1} \text{ и } x''_{i+1} = x'_i, \text{ если } f(x_{i+1}) < 0. \end{aligned} \quad (\text{П.2.5})$$

Здесь и в предыдущей формуле вместо $f(x)$ следует использовать ее точечную оценку по (П.2.3). Метод хорд сходится и для невыпуклых функций $Q(x)$.

3. Метод дихотомии позволяет определять оценки корня $f(x)$ (а следовательно, минимума $Q(x)$) путем деления пополам отрезка, внутри которого находится корень. Пусть, как и при методе хорд, концами такого отрезка являются $x'_i < x''_i$, удовлетворяющие (П.2.4). Тогда в соответствии с этим алгоритмом

$$x_{i+1} = 1/2(x'_i + x''_i).$$

Для очередного шага образуется новый отрезок $[x'_{i+1}, x''_{i+1}]$ по правилу (П.2.5) и т. д. Как видно, для этого метода нет необходимости вычислять значение производной $f(x)$, достаточно определить ее знак. Этот метод сходится для любой одноэкстремальной функции $Q(x)$.

Заметим, что при реализации рассмотренных методов для точного определения положения экстремума необходимо базу оценки g уменьшать с ростом i . Действительно, равенство нулю точечной оценки производной (П.2.3) минимизируемой функции, т. е.

$$Q(x_i + g) - Q(x_i - g) = 0,$$

определяет экстремум лишь при $g \rightarrow 0$.

Теперь рассмотрим методы однопараметрической оптимизации, не опирающиеся на определение производной $f(x)$.

4. Метод золотого сечения. Рассмотрим алгоритм этого метода при его работе из исходного состояния x_0 . Сделаем произвольный шаг a и определим значение минимизируемой функции в полученной точке $Q(x_0 + a)$. Если $Q(x_0 + a) < Q(x_0)$, то следующее определение функции производится в точке $x_0 + a(1 + \tau)$, где $\tau = (1 + \sqrt{5})/2 \approx 1.62$ (рис. П.2.1а), в противном случае в точке $x_0 - a\tau$ (рис. П.2.1б). Теперь на оси x имеются три точки (обозначим их $x^1 < x^2 < x^3$) со значениями минимизируемой функции соответственно Q_1, Q_2 и Q_3 , т. е. $Q_j = Q(x^j)$ ($j = 1, 2, 3$). Так, в первом случае: $x^1 = x_0$, $x^2 = x_0 + a$, $x^3 = x_0 + a(1 + \tau)$, а во втором $x^1 = x_0 - a\tau$, $x^2 = x_0$, $x^3 = x_0 + a$.

Здесь возможны две ситуации:

1) минимальное значение функции качества имеет средняя из этих трех точек, т. е.

$$\min_{j=1, 2, 3} Q_j = Q_2. \quad (\text{П.2.6})$$

Очевидно, что при этом минимум функции $Q(x)$ находится между точками x^1 и x^3 . Поэтому следующее значение минимизируемой функции в этом случае нужно вычислять в точке $x^4 = x^1 + x^3 - x^2$.

Из четырех точек x^j ($j = 1, \dots, 4$) одна крайняя отбрасывается так, чтобы из оставшихся в средней точке значение функции $Q(x)$ было бы минимальным. Эти три точки снова обозначим $x^1 < x^2 < x^3$,

Для них имеет место соотношение (11.2.6). Поэтому здесь снова следует применить это же правило. Так уточняется положение экстремума. Критерием останова процесса может служить, как обычно, точность оценки положения экстремума, т. е. разность $x^3 - x^1$;

2) если же $\min_{j=1,2,3} Q_j \neq Q_2$, т. е. минимально или Q_1 , или Q_3 , то добавляем точку

$$x^4 = \begin{cases} x^1 - \tau(x^2 - x^1), & \text{если } \min_{j=1,2,3} Q_j = Q_1, \\ x^3 + \tau(x^3 - x^2), & \text{если } \min_{j=1,2,3} Q_j = Q_3. \end{cases}$$

Из четырех точек $x^1, \dots, x^4$ выбираем три следующим образом. Если наименьшее значение функции в одной из крайних точек, то отбрасываем точку с наибольшим значением (это будет другая край-

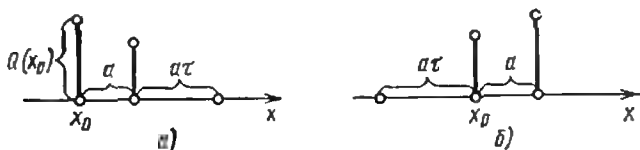


Рис. П.2.1. Расположение точек поиска на начальном этапе поиска (вертикальные отрезки соответствуют значениям функции)

няя точка), а остальные обозначаем через $x^1 < x^2 < x^3$. Эта вторая ситуация и поэтому далее следует обращаться к 2). В противном случае следует отбросить одну из крайних точек так, чтобы из трех оставшихся в средней было бы наименьшее значение функции. Обозначая аналогично эти точки, обнаруживаем, что это первая ситуация, и следует обратиться к 1).

Таким образом, мы приходим к такому значению $x^* = x^2$, которое и является решением задачи оптимизации, где x^2 взято из последней тройки чисел $x^1 < x^2 < x^3$. (Заметим, что в принципе можно взять и иное значение τ , но при указанном τ скорость сходимости процесса оптимизации будет наибольшей).

Изложенные выше методы можно применять лишь при отсутствии случайных помех, накладывающихся на функцию качества $Q(x)$. При наличии таких помех следует обратиться к другим методам поиска экстремума (их алгоритмы часто называют алгоритмами адаптации, подчеркивая этим, что алгоритм поиска как бы приспосабливается, адаптируется к неопределенности, связанной с наличием случайных помех).

5. Метод стохастической аппроксимации используется при минимизации функций, значение которых определяется со случайной помехой. Записывается он в виде

$$x_{i+1} = x_i - \frac{a_{i+1}}{g_{i+1}} [Q(x_i + g_{i+1}) - Q(x_i - g_{i+1})], \quad (\text{П.2.7})$$

Где положительные числа a_i, g_i должны удовлетворять следующим условиям:

$$\lim_{i \rightarrow \infty} a_i = \lim_{i \rightarrow \infty} g_i = 0, \quad \sum_{i=1}^{\infty} a_i = \infty, \quad \sum_{i=1}^{\infty} \left(\frac{a_i}{g_i} \right)^2 < \infty. \quad (\text{П.2.8})$$

Этим условиям (их называют условиями Дворецкого) удовлетворяют, например, последовательности $a_i = 1/i^\alpha$, $g_i = 1/i^\beta$, где на выбор параметров α и β наложены ограничения вида $0 < \alpha < 1$, $\beta > 0$, $\alpha - \beta > 1/2$. В этом случае процесс поиска экстремума функции $Q(x)$ сходится независимо от уровня случайных помех, накладывающихся на эту функцию.

6. *Адаптивный метод* связан с адаптивным изменением длины шага Δx в процессе поиска. Алгоритм заключается в том, что при уменьшении минимизируемой функции делается шаг в том же направлении, а при ее увеличении — в обратном:

$$\Delta x_{i+1} = \begin{cases} \gamma_1 \Delta x_i & \text{если } \Delta Q_i < 0, \\ -\gamma_2 \Delta x_i & \text{если } \Delta Q_i \geq 0, \end{cases} \quad (\text{П.2.9})$$

где $\Delta x_i = x_i - x_{i-1}$; $\Delta Q_i = Q(x_i) - Q(x_{i-1})$, а $\gamma_1 \geq 1$, $0 < \gamma_2 < 1$ — параметры, характеризующие увеличение шага γ_1 при движении в удачном направлении и уменьшение γ_2 при реверсе, т. е. при «переходе» через экстремум. Алгоритм сходится при $\gamma_1 \gamma_2 < 1$.

П.3. Многопараметрический локальный поиск ($n > 1$)

Среди алгоритмов локального поиска большое распространение получили так называемые спусковые алгоритмы, отличающиеся процедурой спуска вдоль выбранного направления. Рассмотрим сначала их. Представим шаг ΔX_i в виде

$$\Delta X_i = a_i V_i, \quad (\text{П.3.1})$$

где V_i — единичный вектор направления на i -м шаге; a_i — величина этого шага. В зависимости от способа формирования вектора V_i образуются различные методы оптимизации.

1. *Покоординатный метод* (Гаусса — Зайделя) характеризуется тем, что направления V_i соответствуют направлениям координатных осей, т. е. $V_i = e_j$ ($j=1, \dots, k$), где e_j — j -й орт, т. е. направление j -й координаты. Следующий шаг будет производиться вдоль $(j+1)$ -й координаты и т. д. до k -й, после чего движение снова будет производиться вдоль первой, т. е. вдоль e_1 , и т. д. Заметим, что здесь и ниже слово «вдоль» означает движение не только в указанном направлении, но и в обратном.

2. *Градиентный метод* (метод наискорейшего спуска) отличается тем, что направление V_i соответствует антиградиентному, т. е.

$$V_i = -\text{dir } \nabla_x Q(X_{i-1}),$$

где dir обозначает направление вектора: $\text{dir } Z = Z/|Z|$, а ∇_x — гра-

диент, т. е. вектор с компонентами, равными частным производным функции, стоящей под этим знаком:

$$\nabla_{\mathbf{X}} Q = \left(\frac{\partial Q}{\partial x_1}, \dots, \frac{\partial Q}{\partial x_k} \right).$$

При этом методе оптимизируемые параметры изменяются так, что минимизируемая функция вначале уменьшается наибо́льшим образом. Для этого, однако, необходимо знать все частные производные этой функции или их оценки. Точечные оценки здесь имеют вид

$$\frac{\partial Q}{\partial x_j} = \frac{1}{g} [Q(\mathbf{X} + g\mathbf{e}_j) - Q(\mathbf{X})], \quad (\text{П.3.2})$$

$$\frac{\partial Q}{\partial x_j} = \frac{1}{2g} [Q(\mathbf{X} + g\mathbf{e}_j) - Q(\mathbf{X} - g\mathbf{e}_j)],$$

где $g > 0$ — база оценки.

Как легко заметить, при $g \rightarrow 0$ обе оценки сходятся к точному значению производной (если она существует, разумеется). Первая оценка в (П.3.2) смещенная, а вторая — несмещенная, но трудоемкость определения последней в два раза больше, так как для нее функцию $Q(\mathbf{X})$ приходится определять в двух состояниях $\mathbf{X} \pm g\mathbf{e}_i$.

3. Метод локального случайного поиска примечателен тем, что направление $\mathbf{V}_i$ является случайным: $\mathbf{V}_i = \Xi$, где Ξ — единичный случайный вектор, равномерно распределенный по всем направлениям пространства параметров $\{\mathbf{X}\}$.

Для реализации каждого из описанных методов поисковой оптимизации необходимо задать величину шага a_i [см. (П.3.1)]. Для определения a_i следует воспользоваться следующим очевидным соображением. Величина $Q(\mathbf{X}_i)$ должна быть возможно малой, т. е. необходимо решить задачу

$$Q(\mathbf{X}_i) = Q(\mathbf{X}_{i-1} + a_i \mathbf{V}_i) \rightarrow \min_{a_i} \Rightarrow a_i^*, \quad (\text{П.3.3})$$

где a_i^* — оптимальная величина шага. Заметим, что, вообще говоря, это может быть и отрицательная величина (например, при случайном поиске). Эта задача однопараметрической оптимизации может решаться стандартными методами одномерного поиска, описанными выше (см. П.2), и обычно вызывается спуском.

4. Метод многомерной стохастической аппроксимации. Алгоритм этого метода записывается в виде

$$\mathbf{X}_{i+1} = \mathbf{X}_i - a_{i+1} \widehat{\nabla}_{\mathbf{X}} Q(\mathbf{X}_i),$$

где $\widehat{\nabla}_{\mathbf{X}} Q(\mathbf{X}_i)$ — точечная оценка градиента, компонентами которой являются оценки частных производных, аналогичные (П.3.2):

$$\frac{\partial Q}{\partial x_j} = \frac{1}{2g_{i+1}} [Q(\mathbf{X}_i + g_{i+1}\mathbf{e}_j) - Q(\mathbf{X}_i - g_{i+1}\mathbf{e}_j)],$$

а переменные коэффициенты a_i и g_i должны удовлетворять условиям Дворецкого (П.2.8).

П.4. Алгоритмы глобального поиска

Если минимизируемая функция $Q(X)$ имеет больше одного экстремума и нужно найти наименьший из них, то ни один из описанных выше алгоритмов локального поиска не решит поставленную задачу. Для этого используют алгоритмы глобального поиска. Рассмотрим некоторые из них.

1. Метод глобального случайного поиска при правильной организации является достаточно эффективным средством отыскания глобального экстремума. Состоит он в следующем. В области поиска Ω в соответствии с заданной, например, нормальной, плотностью распределения $P(X|W, \sigma^2)$ генерируются случайные точки $X_1, X_2, \dots$. Здесь W — математическое ожидание этого распределения, а σ^2 — максимальная дисперсия его компонент. Смысл алгоритма состоит в том, что W располагается в точке с минимальным, найденным ранее значением минимизируемой функции, т. е. $W_i = X^*_i$, где $Q(X^*_i) = Q^*_i = \min_{j=1, \dots, i} Q(X_j)$.

Алгоритм такого поиска записывается в виде рекуррентной формулы для X^*_i и Q^*_i :

$$X^*_i = \begin{cases} X^*_{i-1}, & Q^*_i = Q^*_{i-1}, \text{ если } Q(X_i) \geq Q^*_{i-1}, \\ X_i, & Q^*_i = Q(X_i), \text{ если } Q(X_i) < Q^*_{i-1}. \end{cases}$$

Как легко заметить, этот алгоритм гарантирует отыскание глобального экстремума, если плотность распределения случайных точек не равна нулю ни в одной точке области поиска Ω , т. е.

$$P(X|W, \sigma) \neq 0 \text{ для } X \in \Omega.$$

Действительно, так как вероятность случайного попадания в окрестность глобального экстремума в этом случае конечна, то это событие рано или поздно реализуется. Однако скорее поздно, чем рано, так как сходимость алгоритма медленная, поэтому при реализации этого алгоритма следует принимать меры для увеличения скорости его сходимости.

Для этого применяется адаптация дисперсии σ^2 в процессе поиска: дисперсию уменьшают при неудачных случайных пробах и увеличивают при удачных: $\sigma_i = \sigma_{i-1} + \Delta\sigma_i$, где приращение $\Delta\sigma_i = \varphi(\sigma_{i-1}, \Delta Q_{i-1})$ определяется алгоритмом адаптации. Например,

$$\sigma_i = \begin{cases} \gamma_1 \sigma_{i-1} & \text{при } \Delta Q_{i-1} > 0, \\ \gamma_2 \sigma_{i-1} & \text{при } \Delta Q_{i-1} \leq 0. \end{cases}$$

(Здесь коэффициенты $\gamma_1 < 1$ и $\gamma_2 \geq 1$ определяются из конкретных соображений.) Правильный выбор этих коэффициентов в значительной мере определяет успех глобального поиска. Если они будут близки к единице, то эффект адаптации будет незначительным, а поиск — длительным, но результативным, т. е. будет найден глобальный экстремум. Если же отличие от единицы коэффициентов будет существенным, то процесс закончится быстро, но глобальный экстремум может быть не найден. Чем меньше γ_j , тем быстрее заканчивается поиск и тем больше мы рискуем «проскочить» глобальный экстремум. При выборе γ_1 и γ_2 следует искать разумный компромисс между временем поиска и его надежностью.

2. Метод сглаживающего глобального поиска эффективно применяется в случаях, когда многоэкстремальность минимизируемой функции образуется под влиянием некой колебательной (не обязательно периодической) составляющей $Q_2(X)$, которая накладывается на «хорошую» одноэкстремальную функцию $Q_1(X)$:

$$Q(X) = Q_1(X) + Q_2(X).$$

Априорно это разделение неизвестно. Задача состоит в том, чтобы «сгладить» $Q_2(X)$ и тем самым свести многоэкстремальную задачу к локальной задаче отыскания минимума одноэкстремальной функции $Q_1(X)$, рассмотренной в П.2—П.3.

Процедура сглаживания исходной функции $Q(X)$ должна иметь характер усреднения. Это усреднение производится с учетом весовой функции $p(Y)$, где $Y \in \Omega$ и представляет собой вектор отклонения от X . Весовая функция $p(\cdot)$ определяет вес, с которым происходит усреднение минимизируемой функции $Q(X)$ в окрестности X . Поэтому она должна обладать следующими свойствами:

— неотрицательностью, т. е. $p(Y) \geq 0$ для $Y \in \Omega$;

— нормированностью, т. е. $\int p(Y) dY = 1$;

— δ -образностью по какому-то ее внутреннему параметру q , т. е. $\lim_{q \rightarrow 0} p(Y) = \delta(Y)$, где $\delta(\cdot)$ — дельта-функция.

Гауссова функция

$$p(Y) = \frac{1}{(2\pi)^{n/2} q^n} \exp \left\{ -\frac{1}{2q^2} \sum_{i=1}^k y_i^2 \right\}$$

и многомерный «Локом Ланьезн»

$$p(Y) = l / (1 + q|Y|^2),$$

где l — нормирующий множитель, являются такими функциями.

Теперь процедуру усреднения минимизируемой функции можно представить в виде k -кратного интегрирования с весом p :

$$\bar{Q}(X) = \int_{\Omega} Q_i(X + Y) p(Y) dY, \quad (\text{П.4.1})$$

где $\bar{Q}(X)$ — усредненное в Ω значение функции $Q(X)$.

Точное определение этого интеграла представляет большие трудности. Поэтому его следует оценивать методом Монте-Карло. Весовую функцию $p(Y)$ можно рассматривать как плотность распределения случайного вектора Y . Тогда (П.4.1) запишется в виде

$$\bar{Q}(X) = M_Y Q(X + Y),$$

где операция математического ожидания M_Y берется по случайному вектору Y , имеющему плотность распределения $p(Y)$. Пусть

$Y_1, \dots, Y_N$ — конкретные реализации этого случайного вектора. Тогда получаем монте-карловскую оценку интеграла (П.4.1)

$$\widehat{Q}_N(X) = \frac{1}{N} \sum_{i=1}^N Q(X + Y_i). \quad (\text{П.4.2})$$

Эта оценка обладает очевидным свойством сходимости к точному значению интеграла

$$\lim_{N \rightarrow \infty} \widehat{Q}_N(X) = \bar{Q}(X).$$

Поскольку усредненная функция $\bar{Q}(X)$ получается одноэкстремальной (так как $\bar{Q}(X) \approx Q_1(X)$), то к ней можно применить любые локальные алгоритмы поиска, причем выражение (П.4.2) дает оценку значения этой функции в точке X . Располагая этой оценкой, можно использовать любой из рассмотренных выше алгоритмов локального поиска, т. е. решить многоэкстремальную задачу локальным методом. Однако следует иметь в виду, что каждая реализация оценки сглаженной таким образом функции представляет собой случайную величину, лишь среднее значение которой равно $\bar{Q}(X)$. Оперирование оценками $\widehat{Q}$ в процессе поиска эквивалентно поиску при наличии случайных помех, накладывающихся на функцию $\bar{Q}(X)$. При этом уровень этих помех обратно пропорционален объему выборки N [см. (П.4.2)], что позволяет управлять помехой, точнее уменьшать ее дисперсию путем увеличения N .

Параметр q определяет степень «размытости» весовой функции. Чем больше q , тем сильнее усреднение и «грубее» решение задачи. При малом q усреднение мало. Очевидно, что этот параметр следует изменять в процессе поиска. Вначале он должен быть большим и стремиться к нулю в конце поиска. При хорошем выборе функции $q(i)$ процесс поиска к глобальному экстремуму сходится достаточно быстро.

П.5. Условная оптимизация

Задача условной оптимизации затруднена тем, что ее решение X^* лежит на границе $\bar{\Omega}$ допустимой области Ω , т. е. $X^* \in \bar{\Omega}$. Преодолевают эту трудность двояко. С одной стороны, можно воспользоваться спецификой области Ω , т. е. учесть вид функций, ее определяющих. Этот путь используют методы математического программирования. Некоторые из них рассмотрены в § 6.4. Однако возможен и в ряде случаев более целесообразен другой подход, который заключается в сведении условной задачи оптимизации к безусловной. При этом алгоритм поиска остается прежний (см. П.2—П.4), а основное внимание уделяется процедуре преобразования задачи в безусловную.

Рассмотрим основные подходы в этом направлении.

1. *Метод подстановки.* Он используется при наличии ограничительного типа равенств

$$\Omega : g_i(X) = 0, \quad i=1, \dots, p < k. \quad (\text{П.5.1})$$

Эти ограничения представляют собой систему из p уравнений с k неизвестными. Разрешим систему относительно p переменных через

остальные $k-p$ и подставим эти выражения в $Q(X)$. Размерность решаемой задачи понизится на p , и она становится безусловной. Так и поступают в случае, когда это сделать легко, например для задач линейного программирования типа (6.2.1), которые в результате такого преобразования приводятся к виду (6.4.1).

Очевидно, что подобное преобразование осуществимо только при аналитической разрешимости системы уравнений (П.5.1). Если вспомнить, что в общем случае это могут быть трансцендентные уравнения, то очевидна ограниченность такого подхода.

Однако рекуррентный метод решения задачи оптимизации позволяет обойти эту трудность следующим образом. Разобьем вектор искомых параметров X на два:

$$X = (X^1, X^2), \quad (\text{П.5.2})$$

где

$$X^1 = (x_1, \dots, x_{k-p}); \quad X^2 = (x_{k-p+1}, \dots, x_k), \quad (\text{П.5.3})$$

и будем минимизировать функцию качества с помощью X^1 , а условиям (П.5.1) удовлетворять путем соответствующего подбора X^2 . Последнее сделать просто: достаточно решить систему из p уравнений

$$g_i(X^1, X^2) = 0, \quad i = 1, \dots, p \quad (\text{П.5.4})$$

относительно p неизвестных X^2 при заданных значениях X^1 . Это можно сделать любым численным методом (см. комментарий к П.5). Таким образом, для любого заданного X^1 всегда можно определить такое X^2 , при котором условия (П.5.1) для Ω будут выполнены.

Тогда исходная задача условной оптимизации трансформируется в задачу безусловной оптимизации:

$$Q(X^1, X^2) \rightarrow \min_{X^1},$$

где вектор X^2 определяется на каждом шаге безусловной оптимизации как численное решение этой системы для заданного значения X^1 .

Применим этот подход и для случая, когда имеются ограничения типа неравенств

$$\Omega: h_j(X) \geq 0, \quad j = 1, \dots, q.$$

В этом случае система решаемых уравнений образуется из уравнений нарушенных неравенств

$$h_j(X^1, X^2) = 0, \quad j = z_1, \dots, z_\tau, \quad (\text{П.5.5})$$

где $z_1, \dots, z_\tau$ — номера невыполняющихся ограничений — неравенств

Здесь размерность определяемого вектора X^2 равна τ — числу нарушенных ограничений. В процессе решения системы (П.5.5) число τ может изменяться. Вместе с ним изменяется размерность X^2 . Процесс заканчивается при $\tau = 0$.

Этот подход довольно просто обобщается на случай нарушения ограничений двух типов:

$$\Omega: \begin{cases} g_i(X) = 0, & i = 1, \dots, p, \\ h_j(X_j) \geq 0, & j = 1, \dots, q, \end{cases}$$

где система численно решаемых уравнений образуется из всех нарушенных ограничений.

2. *Метод проекций.* Смысл этого метода сводится к следующему. Если в процессе решения задачи (6.1.8) получено промежуточное значение $X_{N+1} \notin \Omega$, т. е. нарушающее ограничения Ω , то его следует спроецировать на Ω , т. е. удовлетворить ограничениям, и продолжать решение задачи уже из спроецированной точки

$$X^0_{N+1} = \text{пр}_{\Omega} X_{N+1} \in \Omega, \quad (\text{П.5.6})$$

где знак пр_{Ω} означает «проекция на Ω ». Процедура спроецирования наглядно иллюстрируется рис. П.5.1. Если $\Omega: g_i(X)=0$, то процедура проецирования представляет собой спуск на ограничение $g_i(X)=0$ (см. рис. П.5.1,а). Если же область Ω определяется неравенствами (рис. П.5.1,б), то проецирование приводит к одной из точек $X^0 \in \Omega$ этой области.

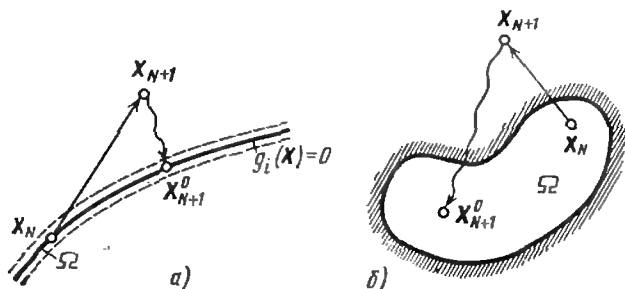


Рис. П.5.1. Иллюстрация работы метода проекций при ограничениях типа равенства (а) и неравенств (б)

Рассмотрим сначала случай, когда $\Omega: g_i(X)=0$ ($i=1, \dots, p < k$). Алгоритм проецирования из точки X на Ω заключается в построении процесса, который бы заканчивался в достаточно малой ϵ -окрестности области Ω , т. е. при

$$|g_i(X^0)| \leq \epsilon_i, \quad i=1, \dots, p \quad (\text{П.5.7})$$

(см. штриховые линии на рис. П.5.1,а), где ϵ_i — заданная точность выполнения i -го ограничения. Этот процесс можно организовать любым методом локального поиска из описанных в П.2, П.3. Для этого достаточно минимизировать функцию качества вида

$$Q^0(X) = \sum_{i=1}^N |g_i(X)| \rightarrow \min_X \quad (\text{П.5.8})$$

при начальных условиях $X=X_{N+1}$ (рис. П.5.1,а) с остановом по условию (П.5.7). Можно решать и другую задачу:

$$Q^0(X) = \max_{i=1, \dots, p} |g_i(X)| \rightarrow \min_X \quad (\text{П.5.8a})$$

с тем же критерием останова.

Аналогично можно поступить и при нарушении ограничений типа неравенств (см. рис. П.5.1,б). В этом случае максимизируется функция качества вида

$$Q^0(X) = - \sum_{j \in \{v\}} h_j(X) \rightarrow \max_X, \quad (\text{П.5.9})$$

где $\{v\}$ — множество номеров нарушенных ограничений, для которых $h_j(X) < 0$. Критерий останова: $Q^0(X) \leq 0$. Аналогично

$$Q^0(X) = - \min_{j=1, \dots, k} h_j(X) \rightarrow \min_X \quad (\text{П.5.9a})$$

с тем же критерием останова. Как видно, определение проекции X^0 точки X удобно производить путем минимизации степени нарушения ограничений, что может быть реализовано с помощью поисковой оптимизации специально организованных функций. Таким образом, при использовании метода проекций для решения условной задачи последовательно шаг за шагом решаются две безусловные задачи, которые можно объединить в одну:

$$Q_*(X) \rightarrow \min_X,$$

где

$$Q_*(X) = \begin{cases} Q(X) & \text{при } X \in \Omega, \\ Q^0(X) & \text{при } X \notin \Omega, \end{cases} \quad (\text{П.5.10})$$

т. е. происходит переключение с одной минимизируемой функции качества на другую в зависимости от выполнения условий Ω .

3. *Метод штрафных функций.* Этот метод является прямым развитием метода проекций, но в нем процесс проецирования совмещен с минимизацией исходной функции. Идея метода заключается в построении такой функции качества, минимум которой представляет собой решение поставленной задачи. Эта функция в общем случае имеет вид

$$\tilde{Q}(X) = Q(X) + f_A(g_1(X), \dots, g_p(X)) + f_B(h_1(X), \dots, h_q(X)), \quad (\text{П.5.11})$$

где $f_A(\cdot)$ и $f_B(\cdot)$ — неотрицательные функции штрафа за нарушение ограничений типа равенств и неравенств соответственно, которые построены следующим образом.

$$f_C = \begin{cases} 0 & \text{при } X \in \Omega, \\ > 0 & \text{при } X \notin \Omega, \end{cases} \quad (C = A, B), \quad (\text{П.5.12})$$

т. е. при выполнении условия Ω штрафа нет, а при невыполнении вводится положительная функция штрафа.

Конструкция функций f_A и f_B может быть весьма разнообразной. Например, штраф за нарушение ограничений типа равенств может быть записан в виде

$$f_A = \sum_{i=1}^p a_i \varphi(g_i(X)), \quad (\text{П.5.13})$$

где $\varphi(\cdot)$ — четная функция со следующими свойствами: $\varphi(x) \geq 0$, $\varphi(0)=0$, $\varphi(x)$ при $x > 0$ — монотонно возрастающая функция. Эти свойства обладает, к примеру, такая функция $\varphi(x) = |x|^\nu$, где $\nu > 0$. Частный случай такой функции штрафа при $a_i=1$ и $\nu=1$ представлен выше (П.5.8).

Функция $f_A = \max_{i=1, \dots, p} a_i \varphi(g_i(X))$ обобщает функцию (П.5.8a).

Как видно, разница между методами проекций и штрафных функций заключается в том, что во втором методе учета ограничений минимизация штрафа и минимизация основной функции происходят одновременно, в то время как в первом эти две процедуры разнесены во времени и происходят независимо друг от друга.

Коэффициенты a_i в функции штрафа (П.5.13) обычно называют коэффициентами штрафа. При $a \rightarrow \infty$, где $a = \min_{i=1, \dots, p} a_i$, решение без-

условной задачи

$$Q(X) + f_A(X) \rightarrow \min_X \Rightarrow X^*_A$$

совпадает с решением исходной условной задачи $Q(X) \rightarrow \min_{X \in \Omega} \Rightarrow X^*$,

где $\Omega: g_i(X)=0$ ($i=1, \dots, p$). Это, очевидно, следует из того, что с ростом величины a минимизация прежде всего коснется функции штрафов, т. е. выполнится условие $X \in \Omega$, и уж после этого будет минимизироваться функция $Q(X)$. Таким образом,

$$\lim_{a \rightarrow \infty} X^*_A = X^*.$$

Теперь рассмотрим ограничения типа неравенств. Функции штрафа в этом случае имеют, например, следующий вид:

$$f_B(X) = \sum_{j=1}^q b_j \psi(h_j(X)), \quad (\text{П.5.14})$$

где $\psi(\cdot)$ — функция вида

$$\psi(z) = -\min(0, z), \quad (\text{П.5.15})$$

$$\text{а } \min(0, z) = \begin{cases} 0 & \text{при } z \geq 0, \\ z & \text{при } z < 0. \end{cases}$$

Выражение (П.5.15) является положительной функцией штрафа за нарушение ограничения $z > 0$. Причем величина этого штрафа тем больше, чем сильнее нарушается ограничение z . Коэффициенты b_j играют роль коэффициентов штрафа.

Другой вид функции штрафа

$$f_B(X) = \max_{j=1, \dots, q} b_j \psi(h_j(X))$$

аналогичен (П.5.9a). При $b \rightarrow \infty$, где $b = \max_{j=1, \dots, q} b_j$, решение безусловной задачи

$$Q(X) + f_B(X) \rightarrow \min_X \Rightarrow X^*_B$$

совпадает с решением условной, где $\Omega : h_j(X) \geq 0$ ($j=1, \dots, q$), т. е. $\lim_{b \rightarrow \infty} X^*_{AB} = X^*$.

Теперь, объединяя оба вида штрафных функций, получаем безусловную задачу вида

$$\tilde{Q}(X) = Q(X) + \sum_{l=1}^p a_l \varphi(g_l(X)) + \sum_{j=1}^q b_j \psi(h_j(X)) \rightarrow \min_X. \quad (\text{П.5.16})$$

Пусть минимум этой функции расположен в точке X^*_{AB} . В общем случае $X^*_{AB} \neq X^*$. Однако при $a, b \rightarrow \infty$, где a и b определены выше, решение безусловной задачи (П.5.16) совпадает с решением исходной задачи, т. е.

$$\lim_{a, b \rightarrow \infty} X^*_{AB} = X^*. \quad (\text{П.5.17})$$

Это обстоятельство и гарантирует успех методу штрафных функций.

Несколько слов об организации процесса решения этим методом. Для того чтобы воспользоваться свойством (П.5.17), необходимо начать оптимизацию при малых значениях параметров a и b , постепенно их увеличивая в процессе поиска до максимальных значений $a_{\max}$ и $b_{\max}$. При этом большую роль играет скорость нарастания этих параметров. Она должна быть оптимальной, т. е. не слишком большой, но и не маленькой.

Действительно, при малой скорости нарастания a и b переход к $a_{\max}$ и $b_{\max}$ будет слишком длительным. Однако при очень большой скорости нарастания a и b функция $\tilde{Q}(X)$ становится овражной (см. рис. 6.3.2, в), что затрудняет ее оптимизацию и удлиняет поиск. Лучшие всего величины a и b изменять адаптивно, т. е. увеличивать в момент, когда имеется уверенность, что найдено решение, достаточно близкое к X^*_{AB} .

П.6. Многокритериальная оптимизация

До сих пор рассматривались методы решения задачи оптимизации скалярной функцией качества $Q(X)$. Однако на практике при оптимизации обычно требуется решение задач, в которых критерий $Q(X)$ является не скаляром, а вектором

$$Q(X) = (q_1(X), \dots, q_m(X))$$

и задача состоит в одновременной минимизации всех m критериев

$$q_l(X) \rightarrow \min_{X \in \Omega}, \quad l = 1, \dots, m, \quad (\text{П.6.1})$$

где Ω — множество допустимых решений, то же, что и в задаче (П.1.1).

На первый взгляд многокритериальная задача кажется неразрешимой. Действительно, решая независимо каждую из m задач

скалярной оптимизации (П.6.1), получаем m оптимальных локальных решений:

$$X^*_{i}: q_i(X^*_{i}) = \min_{X \in \Omega} q_i(X), \quad i = 1, \dots, m.$$

Как легко заметить, эти решения в общем случае не совпадают, т. е. $X^*_{i} \neq X^*_{j}$ ($i \neq j$), и, следовательно, ни одно из локальных решений не может служить решением исходной многокритериальной задачи.

Введем довольно очевидные понятия предпочтения и эквивалентности. Состояние X_1 будем называть строго более предпочтительным, чем состояние X_2 , если $Q(X_1) \leq Q(X_2)$, т. е. среди равенств и неравенств $q_i(X_1) \leq q_i(X_2)$ ($i=1, \dots, m$) всегда найдется хотя бы одно строгое неравенство. Будем этот факт записывать в виде $X_1 > X_2$, если $Q(X_1) \leq Q(X_2)$.

Если же $Q(X_1) = Q(X_2)$, то состояния X_1 и X_2 будем считать эквивалентными: $X_1 \sim X_2$. Заметим, что эквивалентность, вообще говоря, не соответствует равенству. В частном случае, когда $m=k$, может оказаться, что эквивалентность соответствует равенству $X_1 = X_2$, но в общем случае это не так.

Знаки предпочтения ($>$) и эквивалентности ($\sim$) имеют следующие два очевидных свойства:

- 1) если $X_1 > X_2$ и $X_2 > X_3$, то $X_1 > X_3$ — транзитивное свойство предпочтения;
- 2) если $X_1 \sim X_2$ и $X_2 \sim X_3$, то $X_1 \sim X_3$ — транзитивное свойство эквивалентности.

Теперь введем понятие «пространства критериев» $\{Q\}$. Это пространство имеет размерность m (по числу критериев) и образуется m ортогональными осями, вдоль которых откладываются значения $q_i = q_i(X)$. Между пространством параметров $\{X\}$ и пространством критериев $\{Q\}$ имеется тесная связь. Каждая точка пространства параметров X отражается в определенную точку Q пространства критериев (обратное не всегда имеет место).

Область Ω допустимых состояний X в пространстве $\{X\}$, таким образом, отражается в определенную область S пространства критериев $\{Q\}$. На рис. П.6.1 хорошо видно, что любая точка заштрихованного множества более предпочтительна, чем точка $Q(X_0)$ (напомним, что рассматривается задача минимизации). Это обстоятельство позволяет выделить из всего множества S подмножество S^* точек $Q^* \in S^*$, для которых уже не найдется более предпочтительных (см. жирную линию на рис. П.6.1). Однако это вовсе не означает, что множество S^* содержит только лучшие точки. Среди $Q \notin S^*$, вообще говоря, могут оказаться, и даже наверняка имеются точки, в каком-то смысле лучшие, чем некоторые из S^* , но зато среди S^* всегда найдутся точки более предпочтительные, чем $Q \notin S^*$.

Множество $\Omega^* \in \{X\}$, которое соответствует S^* , обычно называют множеством Парето, переговорным множеством или областью

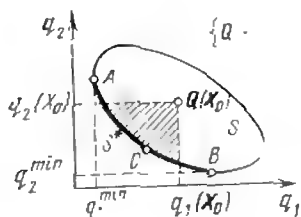


Рис. П.6.1. К определению множества Парето

компромисса. Важность этого множества заключается в том, что именно оно содержит решение многокритериальной задачи (П.6.1).

Пример. Пусть $k=m=2$,

$$q_1 = x_1^2 + x_2^2, \quad q_2 = (x_1 - 1)^2 + (x_2 - 1)^2. \quad (*)$$

$$\Omega: -1 \leq x_i \leq 1 \quad (i=1, 2).$$

На рис. П.6.2,а показана область Ω и множество Парето Ω^* — это отрезок прямой между точками минимума обеих функций (*). Область S для этого примера показана на рис. П.6.2,б, где жирная линия соответствует множеству S^* .

Таким образом, с формальной точки зрения множество Парето следует считать решением многокритериальной задачи (П.6.1). Од-

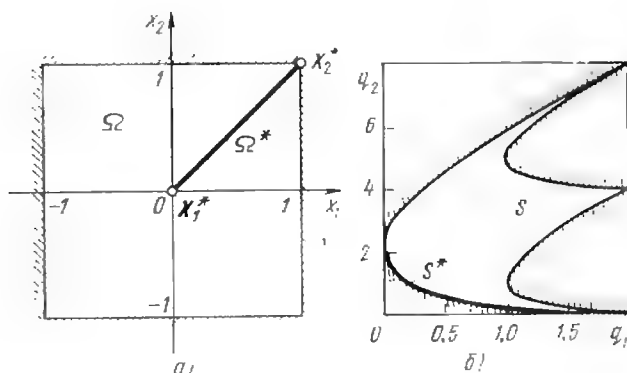


Рис. П.6.2 Пример

нако это «решение» по существу никоим образом не может удовлетворить. Прежде всего хотя бы потому, что это множество решений, а нужно лишь одно. На первый взгляд кажется, что указанное обстоятельство расширяет возможности оптимизации. Действительно, всегда можно предложить еще один дополнительный критерий $q_{m+1}(X)$ и, решив задачу оптимизации на множестве Парето

$$q_{m+1}(X) \rightarrow \min, \\ X \in \Omega^*$$

получить оптимум X^* , удовлетворяющий уже $(m+1)$ -у критерию. Но это ошибочное заключение, так как оптимум получен ценой значительного отклонения от оптимума исходных критериев. Это хорошо видно на рис. П.6.1, где A — положение оптимума по первому (и только первому) критерию. Если же ввести в рассмотрение второй критерий q_2 , то оптимальная точка (пусть она расположена в S) будет значительно отличаться от оптимумов A и B , где B — оптимум по второму критерию q_2 . Таким образом, «свобода», которую представляет множество Парето «куплена» ценой значительного отклонения от локальных оптимумов.

Для однозначного решения многокритериальной задачи оптимизации всегда необходимо введение дополнительной информации I . При отсутствии такой информации о решении X^* ничего

нельзя сказать кроме того, что оно лежит на множестве Парето S^* . II только. Поэтому процесс решения задачи состоит, по сути дела, в поиске и использовании дополнительной информации I , с помощью которой многокритериальная задача сводится к однокритериальной:

$$q(X) = W(q_1(X), \dots, q_m(X) | I), \quad (\text{П.6.2})$$

где $q(X)$ — скалярная функция, минимум которой на Ω^* является решением исходной многокритериальной задачи; W — свертка функции $q_i(X)$ ($i=1, \dots, m$), использующая необходимую информацию I . Простейшим способом сведения многокритериальной задачи к однокритериальной является ее представление в виде стандартной задачи однокритериального математического программирования:

$$q_l(X) \rightarrow \min_{X \in L},$$

$$L: \begin{cases} X \in \Omega, \\ q_i(X) \leq \tilde{q}_i, \quad i \neq l = 1, \dots, m. \end{cases}$$

где информация I заключается в утверждении, что минимизировать надо l -й критерий, а остальные $m-1$ ограничить сверху числами $\tilde{q}_i$ ($i=1, \dots, m$). Так обычно и решались многокритериальные задачи в недалеком прошлом. Но недостаток такого метода очевиден, так как трудно определить «главный» критерий $q_l(X)$ и числа $\tilde{q}_i$ ограничений. Это и заставляет обращаться к другим методам.

Таким образом, различные методы решения многокритериальной задачи отличаются видом и способом получения информации I , а также алгоритмом W «сворачивания» исходных критериев в один глобальный критерий $q(X)$.

Все имеющиеся методы можно подразделить на три класса: априорные, апостериорные и адаптивные методы решения задачи многокритериальной оптимизации. Рассмотрим их.

Априорный метод связан с построением критерия (П.6.2) на основе только априорных сведений об объекте. Например,

$$q(X) = \sum_{i=1}^m \lambda_i q_i(X); \quad q(X) = \prod_{i=1}^m q_i^{\lambda_i}(X)$$

и т. д., где информация I определяет значения коэффициентов $\lambda_1, \dots, \lambda_m$.

«Сворачивание» критериев может производиться и иначе:

$$q(X) = \rho[Q(X) - Q^*],$$

где $\rho(\cdot)$ — некоторая весовая функция расстояния ($\rho(0)=0$, $\rho(\cdot) \geq 0$); Q^* — вектор минимальных значений критериев:

$$Q^* = (q^*_1, \dots, q^*_m), \quad (\text{П.6.3})$$

где $q^*_i = \min_{X \in \Omega} q_i(X)$ ($i=1, \dots, m$).

Здесь решение однозначно определяется заданием весовой функции $\rho(\cdot)$. Например

$$q(X) = \sum_{i=1}^m (q_i(X) - q^*_i)^2.$$

Можно определять решение X^* многокритериальной задачи, исходя из условия совпадения относительных «убытков» по каждому из критериев:

$$\frac{q_i(X) - q_i^*}{q_i^*} = \frac{q_j(X) - q_j^*}{q_j^*}, \quad i, j = 1, \dots, m.$$

Априорный метод следует использовать в том случае, если объем априорной информации достаточно велик, чтобы однозначно выбрать способ «свертывания» всех m критериев в один (П.6.2). Если это не удастся, то обращаются к *апостериорному* методу.

Этот метод связан с организацией сбора информации I до решения поставленной задачи. Для этого прежде всего определим, какими свойствами должна обладать функция свертки $W(\cdot)$ (П.6.2):

1. Функция $W(\cdot)$ должна быть инвариантна по отношению к преобразованию сдвига, т. е.

$$W(q_1, \dots, q_m) = W(q_1 + c_1, \dots, q_m + c_m),$$

где $c_1, \dots, c_m$ — любые числа. Действительно, от добавления к любому из критериев постоянной составляющей решение многокритериальной задачи не должно изменяться. (Это свойство называют аддитивным.)

2. Функция $W(\cdot)$ должна быть инвариантна к изменению масштаба любого критерия, т. е.

$$W(q_1, \dots, q_m) = W(d_1 q_1, \dots, d_m q_m),$$

где d_i — любые положительные числа. (Это свойство называют коммутативным.)

3. Функция $W(\cdot)$ должна быть симметричной относительно всех критериев, т. е. иметь одинаковую структуру по $i=1, \dots, m$. Всем этим требованиям удовлетворяет функция

$$W(X) = \sum_{i=1}^m \alpha_i \frac{q_i(X) - q_i^*}{q_i^{**} - q_i^*}, \quad (\text{П.6.4})$$

где $q_i^{**} = \max_{X \in Q} q_i(X)$, а q_i^* определены выше [см. (П.6.3)]. В этом

легко убедиться при помощи простой подстановки. Каждое слагаемое в (П.6.4) представляет собой не что иное, как нормированный критерий, который может изменяться от 0 до 1.

Теперь рассмотрим способы определения весов α_i ($i=1, \dots, m$). Прежде всего естественно ограничить свободу выбора этих параметров очевидными условиями

$$\sum_{i=1}^m \alpha_i = 1, \quad \alpha_i > 0, \quad i = 1, \dots, m. \quad (\text{П.6.5})$$

Для определения параметров α_i необходимо получить информацию I , которую дает экспертный опрос, например, следующим образом. Экспертам для оценки предпочтения предъявляются несколько пар ситуаций $X_i^{(1)}$ и $X_i^{(2)}$ ($i=1, \dots, p$) со значениями всех критериев в этих точках, т. е. $q_j(X_i^{(1)})$ и $q_j(X_i^{(2)})$ ($j=1, \dots, m$; $i=$

$= 1, \dots, p)$. Эксперты этим парам ставят в соответствие один из трех знаков $>$, $<$ или $\sim$. Это и есть необходимая информация I , с учетом которой имеет место одно из трех соотношений:

$$W(X_i^{(1)}) \leq W(X_i^{(2)}) \quad (i=1, \dots, p).$$

В результате получаем вместе с (П.6.5) систему линейных неравенств и равенств относительно неизвестных весов α_i , разрешая которую стандартными методами, нетрудно найти зону допустимых значений α_i ($i=1, \dots, m$). Теперь осталось выбрать в этой зоне одну точку, например соответствующую центру тяжести этой зоны, и задача определения функции $W(\cdot)$ решена. Далее ее следует минимизировать одним из методов, рассмотренных выше.

Однако в таких условиях эксперту работать не всегда легко. При произвольных ситуациях, предъявляемых ему, он будет чаще отказываться от решения, чем принимать его. Именно в этих случаях следует обратиться к *адаптивному* методу.

Такой метод опирается на использование экспертных решений в процессе решения многокритериальной задачи. Суть его сводится к следующему. Эксперту предъявляются какая-то i -я ситуация $X_i \in \Omega^*$ (на Парето) и указываются численные значения всех критериев в этой ситуации: $q_1(X_i), \dots, q_m(X_i)$. Эксперт принимает решение, по каким из m критериев следует сделать шаг минимизации (в ущерб остальным, разумеется). Эту процедуру можно выполнить формально, переходя в точку $X_{i+1} \in S^*$, обладающую свойствами

$$q_{k_j}(X_i) < q_{k_j}(X_{i-1}), \quad (\text{П.6.6})$$

где k_j ($j=1, \dots, k^i < m$) — номера критериев, указанные экспертом на i -м шаге (их k^i штук). Это и есть информация I , поступившая на i -м шаге. Решить задачу (П.6.6) нетрудно, тем более, что точка X_{i+1} должна быть расположена недалеко от X_i . В полученной точке X_{i+1} определяются все значения критериев, которые снова предъявляются эксперту, и т. д. до тех пор, пока эксперт не остановится на решении X^* , которое он признает оптимальным. Это и есть решение многокритериальной задачи.

ЛИТЕРАТУРНЫЙ КОММЕНТАРИЙ К ПРИЛОЖЕНИЮ

Задача оптимизации, сформулированная в П.1, является по сути дела задачей математического программирования с той разницей, что минимизируемая функция и функции ограничений могут быть заданы алгоритмически, т. е. правилом вычисления их значений, а не явно аналитически, как в математическом программировании. Это обстоятельство не позволяет воспользоваться информацией о структуре указанных функций в процессе поиска решения, что заставляет обращаться к достаточно универсальным алгоритмам поиска.

Приведенная здесь классификация очень условна (как и все классификации) и имеет целью лишь ориентировать читателя в структуре приложения. Более полный обзор можно найти в книге [1].

Для первого ознакомления с методами поисковой оптимизации читателю рекомендуется научно-популярная книга [2].

Одномерный поиск, рассмотренный в П.2, здесь не является самостоятельным алгоритмом, так как слишком мало задач управления сводятся к оптимизации по одной переменной. Он входит в качестве процедуры спуска вдоль выбранного направления при многопараметрической оптимизации. Методов одномерной оптимизации много. Наиболее полно они изложены в [3].

В П.3 описаны в основном лишь спусковые алгоритмы многопараметрической оптимизации. Однако это лишь незначительная часть существующих методов. Их подробное изложение можно найти в [1—8]. Кроме того, для решения оптимизационных задач можно воспользоваться некоторыми методами нелинейного математического программирования, не слишком сильно связанными с характером нелинейности (см. список литературы к гл. 6).

Поиск глобального экстремума, рассмотренный в П.4, является, по-видимому, самой сложной задачей оптимизации. Для ее решения привлекаются разнообразные алгоритмы случайного поиска [1, 4, 6, 7, 9—11]. Имеются и детерминированные алгоритмы глобального поиска [12, 13].

Идея сглаживания с использованием метода Монте-Карло дала возможность сконструировать весьма эффективные алгоритмы глобального поиска [14, 15].

Наиболее распространенным методом решения задач условной оптимизации является метод штрафных функций, описанный в П.5. Более подробно с ним можно познакомиться по [8, 16].

Задачи многокритериальной векторной оптимизации, рассмотренные в П.6, являются достаточно новым классом задач, и основной проблемой здесь является способ сворачивания всех критериев в один глобальный. Различные способы такой свертки рассмотрены в [17—19]. Популярное изложение проблемы принятия оптимальных решений в многокритериальной ситуации можно найти в брошюре [20]. Обзоры [21, 22] и сборник [23] позволяют получить полное представление о существующих методах многокритериальной оптимизации.

СПИСОК ЛИТЕРАТУРЫ

1. Растринин Л. А. Системы экстремального управления. М.: Наука, 1974.
2. Первозванский А. А. Поиск. — М.: Наука.
3. Уайлд Д. Дж. Методы поиска экстремума: Пер. с англ. — М.: Наука, 1967.
4. Растринин Л. А. Статистические методы поиска. — М.: Наука, 1968.
5. Пшеничный Б. Н., Данилин Ю. М. Численные методы в экстремальных задачах. — М.: Наука, 1975.
6. Гурии Л. С., Дымарский Я. С., Меркулов А. Д. Задачи и методы оптимального распределения ресурсов. — М.: Сов. радио, 1968.
7. Алгоритмы оптимизации проектных решений/ Под ред. А. И. Половинкина. — М.: Энергия.
8. Химмельблау Д. Прикладное нелинейное программирование: Пер. с англ. — М.: Мир, 1975.
9. Моцкус И. Б. Многоэкстремальные задачи в проектировании. — М.: Наука, 1967.

10. Юдин Д. Б. Математические методы управления в условиях неполной информации. — М.: Сов. радио, 1974.
11. Коган Б. М., Тер-Микаэлян Т. М. Решение инженерных задач на ЦВМ. — М.: Энергия, 1964.
12. Уланов Г. М., Носудлаев Н. Н. Многоэкстремальные системы управления — В кн.: Техническая кибернетика/ ВШНТИ. — М., 1972. - Т. 4.
13. Стронин Р. Г. Информационный метод многоэкстремальной оптимизации при измерениях с помехами. — Изв. АН СССР, Техническая кибернетика. 1966. № 3.
14. Катковник В. Я. Методы алгоритмической оптимизации. — В кн.: Методы исследования нелинейных систем автоматического управления/ Под ред. Р. А. Неллина. — М.: Наука, 1975.
15. Катковник В. Я. Липсные оценки и стохастические задачи оптимизации: метод параметрических операторов усреднения — М.: Наука, 1976.
16. Фиакко А., Мак-Кормик Г. Нелинейное программирование: методы последовательной безусловной минимизации: Пер. с англ.— М.: Мир, 1972.
17. Гермейер Ю. Б. Введение в теорию исследования операций. — М.: Наука, 1971.
18. Салуквадзе М. Е. Задачи векторной оптимизации в теории управления. — Тбилиси: Мецниереба, 1975.
19. Гуткин Л. С. Оптимизация радиотехнических устройств по совокупности показателей качества — М.: Сов. радио, 1975.
20. Солнышков Ю. С. Количественное обоснование решений. — М.: Знание, 1971.
21. Ларичев О. И. Человеко-машинные процедуры принятия решений. — Автоматика и телемеханика, 1971, № 12.
22. Емельянов С. В., Борисов В. И., Молевич А. А., Черкашин А. М. Модели и методы векторной оптимизации. — В кн.: Итоги науки и техники. Техническая кибернетика, № 5. — М.: ВШНТИ, 1973.
23. Многокритериальные задачи принятия решений/ Под ред. Д. М. Гвишiani и С. В. Емельянова. — М.: Машиностроение, 1978.

Именной указатель

- Абчук В. А. 180
 Авдеева Л. П. 180, 181
 Адамс Г. Э. 180
 Адлер Ю. П. 5, 143
 Айзерман М. А. 203, 204
 Акофф Р. 62, 53, 187, 188
 Аркадьев А. Г. 203, 204
 Балакирев С. В. 118
 Басов А. С. 180, 181
 Бендат Дж. 203, 204
 Бесекерский В. А. 187, 188
 Бешелев С. Д. 70, 71
 Бир С. 39, 40
 Бусленко Н. П. 102, 103
 Борисов В. И. 225, 226
 Браверман Э. М. 203, 204
 Бонгард М. М. 203, 204
 Вагнер Г. 102, 103, 180, 181
 Вапник В. Н. 203, 204
 Ватель Н. А. 187, 188
 Винер Н. 9, 39, 40
 Воронов А. А. 187, 188
 Габасов Р. 180
 Гасс С. 180
 Гавишани Д. М. 225, 226
 Гейл Д. 180, 181
 Гейлс У. М. 180
 Гельвеций 3
 Гермейер Ю. Б. 187, 188, 225, 226
 Глушков В. М. 39, 40
 Гольштейн Е. Г. 180, 181
 Граве П. С. 39, 40, 52, 53
 Грановский Ю. В. 141
 де Гроот М. 70, 71
 Гурович Ф. Г. 70, 71
 Гуткил Л. С. 225, 226
 Данилин Ю. М. 224, 225
 Данилов Ф. А. 118
 Даффин Р. 180, 181
 Дворецкий 211
 Денисов В. И. 141
 Джонсон Р. 40
 Дудников Е. Г. 118
 Дымарский Я. С. 225
 Егоров К. В. 187, 188
 Емельянов С. В. 225, 226
 Ерешко Ф. И. 187, 188
 Ермаков С. М. 102
 Затворуйко Н. I 203, 204
 Зайченко Ю. П. 179, 180
 Захаров В. Н. 3
 Зенер К. 180, 181
 Зуховицкий С. И. 180, 181
 Интрилигатор М. 179, 180
 Карлин С. 180, 181
 Карпелевич Ф. Н. 180, 181
 Каст Ф. 40, 39
 Кинг В. 39, 40
 Кириллова Ф. М. 5, 180
 Клитланд Д. 39, 40
 Кориков А. М. 5
 Коршунов Ю. М. 180
 Корявов П. П. 102, 103
 Кофман А. 180
 Коган В. М. 225, 226
 Круг Г. К. 141
 Кузин Л. Т. 179, 180
 Куо Б. 180, 181
 Ларичев О. И. 225, 226
 Лефевр В. А. 187, 188
 Лецкий Э. К. 141
 Ли Т. Г. 180
 Лихтенштейн В. Е. 180, 181
 Ляшенко И. Н. 180
 Мак-Кормик Г. 180, 181
 Маджаров Н. Е. 102, 118
 Мако Д. 52, 53
 Маркова Е. В. 141
 Медич Д. 70, 71
 Месарович М. 52, 53
 Тер-Микаэлян Т. М. 225, 226
 Мозес Л. 70, 71
 Молевич А. А. 225, 226
 Моисеев П. И. 102, 103, 180,
 Мукачева Э. А. 180, 181
 Меркулов А. Д. 225
 Моцкус Н. Б. 225
 Налимов В. В. 140, 141
 Нейлор Т. 102, 103
 Неймарк Ю. 5
 Петушин А. В. 187
 Носовласов П. Н. 225, 226
 Опnier С. J 40
 Ордынцев В. М. 118
 Павлов В. В. 188
 Первозванский А. А. 225

Перегудов Ф. И. 40
 Питерсон Э. 180, 181
 Поляк Ю. Г. 102, 103
 Полетаев И. А. 5
 Подов Е. П. 187, 188
 Поспелов Д. А. 102, 103
 Пугачев В. С. 187, 188, 203
 Пупков К. А. 179, 180
 Половинкин А. И. 225
 Пшеничный Б. Н. 225
 Райбман Н. С. 118
 Раскин Л. Г. 179, 180
 Растринин Л. А. 39, 62, 63, 102, 118, 225
 Розенцвейг Д. 40
 Розоноэр Л. И. 203, 204
 Ромакин М. И. 180, 181
 Рубан А. Н. 5
 Рубинштейн Г. Ш. 180
 Саати Т. 180, 181
 Садовский А. Е. 180, 181
 Салуквадзе М. Е. 225, 226
 Садошников Р. А. 187, 188
 Свешников А. А. 181, 203
 Скляревич Л. 5
 Сковорода Г. 6
 Соболев И. М. 103, 103
 Солнышков Ю. С. 225, 226
 Сосулин Ю. А. 111
 Стронгин Р. Г. 225, 226
 Сушков Б. Г. 102, 103
 Табак А. 180, 181
 Тарасенко В. П. 5

Тарасенко Ф. П. 5
 Тахакара И. 52, 53
 Тихонов В. И. 203, 204
 Уайлд Д. Дж. 224, 225
 Уемов А. И. 70, 71
 Уланов Г. М. 225, 226
 Фатусов В. А. 141
 Федоренко Н. П. 39
 Федоров В. В. 141
 Фнакко А. 180, 181
 Финкельштейн Ю. Ю. 180, 181
 Форрестер Дж. 102, 103
 Хазацкий В. Е. 3
 Химмельблау Д. 141, 225
 Хургин Я. И. 140, 141
 Цирлин А. М. 118
 Чадеев В. М. 118
 Червоненкис А. Я. 203, 204
 Черкашин А. М. 225, 226
 Чернов Г. 70, 71
 Черняк Ю. И. 40
 Шеннон К. 102, 103
 Шляпентох В. 70, 71
 Эйкхофф П. 118
 Эмери Ф. 52, 53
 Эшби У. 39, 40
 Юдин Д. Б. 180, 181, 225, 226
 Юевич Е. И. 187, 188
 Янг С. 40

Предметный указатель

Абсолютно управляемый объект 63
 Адаптация модели 29
 Адаптивная идентификация в задачах управления точностью прокатки труб 118
 Адаптивный алгоритм идентификации 105
 — метод 224
 Аддитивная модель стохастичности 92
 Активность объекта 184, 187
 Алгоритм 115
 Алгоритмы случайного поиска 212
 Априорные и апостериорные данные 104
 Базовые отношения 97
 — понятия 97
 Взаимодействие субъекта с системой управления 44
 — целевой области и траектории изменения среды 51
 Входы и выходы объекта 71
 — —, экспертное ранжирование 74
 Выделение объекта из среды 53
 Выпуклая область 151
 Границы объекта 54
 Декомпозиция модели 85

—, влияние на сложность 89
 — на два элемента 88
 Динамический объект 193
 Динамическое программирование 171
 Динамичность 90
 Дискриминирующие планы 135
 Дисперсионная матрица параметров 125
 D-критерий оптимальности 126
 D-оптимальное планирование 127
 Задача вариационная 145
 — величинного программирования 151
 — оптимизации функций многих переменных 205
 — синтеза управления 179
 — стохастического программирования 152
 — целочисленного программирования 153
 Задачи линейного программирования 148
 — —, пример образования выпуклого многогранника 160
 — математического программирования 148, 155, 179
 — — —, методы решения 158

- оптимального проектирования 161
- синтеза решающего правила 201
- целочисленного программирования 163
- —, методы решения 164
- Знаки предпочтения и эквивалентности 220
- Зоны неуправляемости 67
- планирования в пространстве управляемых входов 123
- Игры неантагонистические** 187
- Идентификация динамических моделей 114
 - параметров модели объекта 21, 103
 - промышленных объектов 118
 - сложных систем 109
 - статических объектов 107
 - динамических моделей 114
 - промышленных объектов 118
- Идея усиления контраста 83
- Имена 98
- Имитационное моделирование 94
- Коррекция объекта управления** 197
 - параметров модели объекта 189, 203
 - системы управления 4, 29, 188
 - целей управления 199
- Кoeffициент компетентности 67
- Критерий планирования 121, 124
 - согласованности экспертов 79
- Локальность задачи выпуклого программирования** 150
- Линии равного уровня 157
- Математическое описание** 18
 - программирование 145
- Матрица ортогональная относительно столбцов 129
 - планирования 121
 - средних предпочтений 84
- Метод адаптивный 210
 - апостериорный 223
 - априорный 222
 - возможных направлений 161
 - глобального случайного поиска 212
 - градиентный 210
 - дихотомии 208
 - золотого сечения 208
 - касательных 207
 - Монте-Карло (статистических испытаний) 96, 102
 - локального случайного поиска 211
 - непосредственного ранжирования 76
 - парных сравнений 81
 - подстановки 214
 - по координатам (Гаусса—Зейделя) 210
 - проекций 216
 - сглаживания глобального поиска 213
 - стохастической аппроксимации 209
- Методы многомерной стохастической аппроксимации 211
 - поисковой оптимизации 205
 - решения вариационных задач 167, 180
- Многокритериальная оптимизация 219
- Множество Парето 220
- Модели имитационные 91, 102
 - семиотические 97, 102
- Моделирование сложных объектов 94
- Модель объекта управления 22
 - субъекта 12
- Невоспроизводимость экспериментов 19
- Некомфортность субъекта 45
- Нелинейность 91
- Неадаптивный алгоритм идентификации 106
- Ненаблюдаемое возмущение 15
- Нестационарность сложного объекта 20, 93
- «Нетерпимость» к управлению 19
- Обратный поиск 166
- Объект планирования 120
 - управления 9, 21, 63
 - —, зоны неуправляемости 65
 - эргатический 187
- Объекты линейные 91
 - нелинейные 92
- Однопараметрический локальный поиск 206
- Оператор 10
- Оптимальное управление 146, 176
- Оптимальные цели управления 46
- Оптимум «размеров» объекта 54
- Парадокс цели 41
- Пассивное наблюдение 25
- Планирование эксперимента 25, 103, 118, 140
 - — последовательное 133
 - —, теория 120
- Правило вычисления рангов по матрице 82
 - транзитивности 82
- Преобразование исходных описаний 101
- Признаки задачи оптимизации 205
- Принцип максимума 167
- Показатели эффективности структур 196
- Пространство ситуаций 47
 - целей 49
- Ранжирование 75
- Реализация управления 181
- Рекуррентный метод 155
- Рельеф функции 157
 - целевых функций и показателей качества 186
- Рельефы функции качества 157
- Ресурсы планирования 121
- Связи объекта со средней 72
- Симметрия 101
- Симплексный метод 159
- Синтаксически правильные фразы 98
- Синтез модели управления 4, 71
 - — — параметрический 103
 - — — структурный 71
 - управления 26, 142
- Система управления 11
 - — для составления коктейля 31
- Системный подход 38
- Ситуация 60

След матрицы 126
Сложная система 18
Сложность 85
Сложный объект управления 17
Согласованность экспертов 69, 78
Старение информации 202
Статистический объект 183, 191
Стохастичность поведения сложных объектов 18
— структуры объекта 92
Структура 23
— задач поисковой оптимизации 206
— модели 89, 102, 119
— объекта динамическая 90
— — статистическая 90
Структурная адаптация 203
Структурный синтез модели 23, 71, 102

Транзитивности правило 82

Управление 6, 8
— помощью формальных методов 18

— динамическим объектом 147
— сложным объектом 20
Управления реализация 4, 28
Управляемость объекта 60
— —, экспертный метод оценки 67
— частичная (относительная) 64
Условия Дворецкого 210
Условная оптимизация 214

Формализм в управлении 6
Формулировка целей управления 21, 50

Функция 10
— выпуклая вниз 150
— Гамильтона 169

Целевые переменные 48
Цель управления 4, 15, 41, 20

«Черный ящик» 72, 85

Экспертная оценка неуправляемости объекта 69

Язык бинарных отношений 97

Оглавление

От автора	3
Предисловие	6
Введение	9
В.1. Что такое управление?	9
В.2. Система управления	14
В.3. Что такое «сложный объект управления»?	17
В.4. Этапы управления сложным объектом	21
В.5. Как сделать коктейль? (Несерьезная иллюстрация серьезных этапов управления)	31
Литературный комментарий к введению	39
Список литературы	40
 Глава 1. Формулировка целей управления	41
1.1. Парадокс цели	41
1.2. Модель субъекта	42
1.3. Пространства ситуаций и целей	46
Литературный комментарий к главе 1	52
Список литературы	53
 Глава 2. Определение объекта управления	53
2.1. Анализ проблемы	53
2.2. Метод экспертных оценок	56
2.3. Управляемость объекта	60
2.4. Экспертный метод оценки управляемости объекта	67
Литературный комментарий к главе 2	70
Список литературы	71
 Глава 3. Структурный синтез модели объекта	71
3.1. Определение входов и выходов объекта	71
3.2. Экспертное ранжирование входов и выходов	74
3.3. Декомпозиция модели	85
3.4. Структура модели	89
3.5. Имитационные модели	94
3.6. Семiotические модели	97
Литературный комментарий к главе 3	102
Список литературы	102
 Глава 4. Идентификация параметров модели	103
4.1. Постановка задачи идентификации	104
4.2. Идентификация статических объектов	107
4.3. Идентификация динамических моделей	114
Литературный комментарий к главе 4	118
Список литературы	118

Глава 5. Планирование экспериментов	118
5.1. Основные понятия теории планирования экспериментов	120
5.2. Критерии планирования	124
5.3. D-оптимальное планирование	127
5.4. Последовательное планирование экспериментов	133
5.5. Дискриминирующие планы	135
<i>Литературный комментарий к главе 5</i>	140
Список литературы	141
 Глава 6. Синтез управления (принятие решения)	 142
6.1. Постановка задачи	142
6.2. Классификация задач математического программирования	148
6.3. Анализ задач математического программирования	155
6.4. Некоторые специальные методы решения задач математического программирования	158
6.5. Некоторые специальные методы решения вариационных задач	167
<i>Литературный комментарий к главе 6</i>	179
Список литературы	180
 Глава 7. Реализация управления	 181
7.1. Постановка задачи	182
7.2. Учет влияния среды	183
7.3. Учет активности объекта	184
<i>Литературный комментарий к главе 7</i>	187
Список литературы	187
 Глава 8. Коррекция системы управления	 188
8.1. Коррекция параметров модели объекта	189
8.2. Коррекция структуры модели	194
8.3. Коррекция объекта управления	197
8.4. Коррекция целей управления	199
<i>Литературный комментарий к главе 8</i>	203
Список литературы	203
 Приложение. Методы поисковой оптимизации	 205
П.1. Постановка задачи	205
П.2. Однопараметрический локальный поиск ($n=1$)	206
П.3. Многопараметрический локальный поиск ($n>1$)	210
П.4. Алгоритмы глобального поиска	212
П.5. Условная оптимизация	214
П.6. Многокритериальная оптимизация	219
<i>Литературный комментарий к приложению</i>	224
Список литературы	225
Именной указатель	227
Предметный указатель	228